REMOVING BARRIERS TO THE USE OF SIMULATION IN THE BUILDING DESIGN PROFESSIONS

Jon William Hand B.Sc., M.Arch.

A thesis submitted for the
Degree of Doctor of Philosophy

Energy Systems Research Unit
Department of Mechanical Engineering
University of Strathclyde, Glasgow, UK.

November, 1998

Table of Contents

COPYRIGHT DECLARATION

ABSTRACT

This thesis is concerned with removing barriers to the use of simulation within the building design professions. It employs a case study approach to identify facets of current generation simulation tools, simulation practice and skills acquisition which are problematic. The case studies include several leading-edge European research initiatives, consulting projects, workshops and teaching initiatives.

These observations are used as the foundation for several cycles of conjecture and testing, culminating in a Project Manager application which controls all aspects of simulation-based design decision support. The design of the Project Manager is founded on the interface between the practitioner and the underlying data model. The Project Manager encompasses: tight binding of the interface to the underlying data model; expression of the data model as objects in the user's domain; the containment of information related to all phases of a design project; an un-noticed interface; access to on-line support and a central desktop tool metaphor. A desktop for visual assessments has also been developed to support integrated thermal and visual assessments. The Project Manager is then tested in the context of a design study of considerable complexity, including integrated performance assessment and the cooperative use of design tools. It demonstrates simulation's support within the constraints of design practice.

The management of simulation projects is then extended by the inclusion of knowledge-based control of the modelling process within an integrated building design system.

The work shows that the efficacy of simulation within the design process is enhanced by:

- the use of progressive exercises in formal skills acquisition, mentor-based training for practitioners and access to a range of exemplar models;
- recognition of the ad hoc and iterative nature of the design process and the designer's need for early confirmation of performance trends;
- extensions to the simulation data model beyond the description of thermophysical and systems details to contain other aspects of the design process;
- tight binding of the tool interface to the underlying data model, cooperative use of graphic views and logically named attribution to enhance the clarity of models;
- the introduction of project management facilities to coordinate all aspects of simulation work and enable the exchange of simulation models between assessment tools;
- the cooperative working of assessment tools to support integrated assessments of designs of realistic complexity and the production of integrated views of performance;
- the introduction of knowledge based control of the modelling process.

ACKNOWLEDGMENTS

This thesis could have been completed only within the exceptional group environment of the Energy Systems Research Unit of the University of Strathclyde. Where else could an architect creatively compose a virtual edifice which addresses so many issues of the design process?

I am indebted to those who have read and commented on the work, Dr. Paul Morgan, who challenged me to think beyond the sounds of words, Brenda Lillicrap whose green pen focused the storyline and Professor Joe Clarke who patiently coerced this missive into a document which delivers useful insights.

I dedicate this to my Parents, who never stopped believing that it could be done.

Chapter 1

THE NATURE OF THE PROBLEM

1 THE NATURE OF THE PROBLEM

> *Assessments of change, dynamics, and cause and effect are at the heart of thinking and explanation. To understand is to know what cause provokes what effects, by what means, at what rate. Edward R. Tufte, Visual Explanations: Images and Quantities, Evidence and Narrative (Cheshire, Connecticut 1997 p9)* .

One of the challenges in the design of the built environment is to understand and harness the interactions between heat and mass transfer and complex control systems in order to make best use of resources, improve comfort and productivity and test the viability of novel designs. While such interactions are substantially known to the engineering and scientific communities, and tools which take them into account are hardly new, most "best practice" analysis eschews observed complexity and dynamics. This constrains the design process and may result in designs which fall short of the expectations of the end user, client or design team.

One causal factor is the difficulty design professionals (e.g. architects, environmental, mechanical and control engineers) have in deploying design decision support tools which are capable of dealing with realistic complexity. This thesis investigates the design and deployment of such tools. It identifies constraints, tests alternative approaches to tool design, examines alternative assessment procedures and modes of skills acquisition.

The imperative for this work derives from several observations as follows.

- Fluctuations in the availability and cost of energy, as well as the related environmental impacts, give rise to questions about the efficiency of environmental systems, and the desire to design buildings that make better use of natural resources.
- Contemporary design decision support must deal with issues such as indoor air quality, human comfort, demand-side energy management and renewable energy systems integration which are beyond the remit of traditional assessment methods.
- An over-reliance on mechanical systems has obscured the inherent thermal and visual performance implications of architectural and engineering design decisions. Many architects are not well versed in either building physics or mechanical systems. At the same time, many engineers view mechanical systems and controls as decoupled from building design. Without clear feedback and

the ability to test the implications of design decisions, only the most conservative of design options will be implemented.

- The pressure for competitive differentiation is leading property developers and designers to include novel design features (such as atria) in their core design work. Unfortunately, there is considerable uncertainty in assessing the performance of such designs if the dynamic and integrated response of the building and its environmental systems is not taken into account.

According to the Energy Design Advisory Service (EDAS) [McElroy 1993], the design professions are increasingly attempting to meld passive and active design strategies in their work (especially within prestige buildings). The Chartered Institute of Building Services Engineers (CIBSE) [Hand et al. 1998] notes the increasing sensitivity of buildings to variations in internal loads and raises concerns over indoor air quality as building envelopes become tighter and new materials are introduced. Clearly, assessing the performance of such designs requires considerable skills within the design team. Both the risk of failure and the goal of achieving an exceptional design argue for access to sophisticated design decision support.

In response, scores of decision support tools have evolved. At one extreme these codify traditional engineering approaches; at the other extreme they attempt to simulate the real world from first principles. An indication of the diversity of tools can be found in a "Tools Directory" Web page hosted by the United States Department of Energy [Department of Energy 1998].

Those wishing to classify such tools often do so by looking at the underlying computational regime and the nature of the design problems which tools support. Clarke and Maver [1991] suggest four generations:

1st generation: Such tools are handbook oriented computer implementations, analytical in formulation, and biased towards simplicity. They are piecemeal in their approach, providing indicative results within constrained solution domains.

2nd generation: Such tools are characterised by the introduction of the dynamics of fabric response, but decoupled in relation to the treatment of air movement, systems and control. Early implementations were decoupled from the design process by limited interfaces and computational requirements which were substantial for their time. Later implementations are often marketed on their ease of use and speed of solution.

3rd generation: Such tools are characterised by treating the entire building as a coupled field problem and employing a mix of numerical and analytical techniques. These tools demand considerable expertise and resources to go beyond simple problems. Interfaces are able to reduce some barriers to their use. Modelling integrity is enhanced but is often used to derive information to be incorporated in simplified techniques. Tools such as NBSLD [Kusuda 1976] and, more recently, TSBI3 [Johnsen and Grau 1994], have been developed in institutions which support standards.

4th generation: Such tools are characterised by full computer- aided building design integration and advanced numerical methods which allow integrated performance assessments across analysis domains. Interfaces and underlying data models are evolved to present and operate on simulation entities as objects in the user's domain. One common evolution is the incorporation of knowledge bases within the tool infrastructure.

In this document, the third and fourth generations are referred to as *simulation*. The first and second generations are referred to as *simplified methods* because of their constrained treatment of the underlying physics. Tools which are focused on a specific assessment domain (e.g. computational fluid dynamics, thermal bridges, glazing system design) may fall into either category, depending on their treatment of the underlying physics.

1.1 Surveys and comments

Others have surveyed the design and simulation community to determine the patterns of use of assessment tools and the barriers to their use. For example, Parand [1996] found that:

> *"...surveys agree on the main barriers to the use of [building energy and environment] software. These can be summarised as follows: perceived steep learning curve, ease of use and user interface, fear of effect of user errors, time required for preparation and calculation, lack of CAD integration, lack of suitable default values, lack of good datasets, credibility of predictions."*

Another view [Selkowitz 1992] is:

> *"The need for even greater advances in energy efficiency in buildings remains, but is hampered by a lack of interest, lack of incentive, and lack of understanding of what is possible and what is required to achieve those advances. Fundamentally, energy efficiency remains low on the priority list. If we are to make continued progress toward efficiency goals we therefore need strategies that link efficiency investments to other more desirable features and services, such as comfort and productivity."*

Selkowitz gives as an example the evolution of lighting design tasks from reading fifth carbon copies to assessing the visibility of computer monitors and the need to understand advanced lighting controls. A need for alternatives to worst case system design approaches is also put forward. He calls on tool designers to deliver support for such design decisions and provide performance information in forms which are clear to the practitioner—such as systems performance via psychrometric charts, or in the case of lighting, alternatives to traditional daylight factors.

Selkowitz also suggests that:

> *"...an ideal building design tool should: a) be interactive with the designer, accommodate different users and skill levels, and match the design process, b) present information in an appropriate format for architects, c) provide efficient access to large databases (tools that*

> *make gigabytes of data available to the designer will not help solve real problem unless the data can be stored, retrieved and manipulated in a reasonable way), d) provide design guidance and expert advice and support for commissioning."*

According to a survey of 69 assessment tool users by the Building Performance Research Unit of Anglia Polytechnic University [Robinson 1996], the required improvements for detailed programs were evenly spread between interface and data management issues, links to CAD, better reporting facilities, application documentation, and more comprehensive databases. The need for more comprehensive training was also mentioned. Two findings, which went against conventional assumptions, were that users of complex tools conducted more iterations of assessments over a wider range of issues and did so earlier in the design process than those who used simplified tools. The fact that simulation supports a wider range of parameter variations and can focus on more issues might be a contributor to the first observations. The nature of simulation does not preclude constrained models and so the latter finding might gain credibility as simulation is deployed in practice.

Although the majority of respondents were engineers, others in the design process make decisions which are based on thermal and visual performance and it has long been a goal of workers in the field to broaden the audience for simulation. A perspective from an architectural point of view [Howrie 1995] emphasised:

- the iterative nature of design and the need for support during early iterations;
- the need for co-ordination and integration of the design process.

Howrie states "Currently, it is my perception that architects and engineers operate virtually discrete processes in designing the same building. The language, the concepts and the sequences of thought of each are barely recognisable one to the other." He also observed that many engineers are detached from the rapid and *ad hoc* iteration experienced by the architect and client.

In terms of the metrics of assessment, Howrie notes the preference of architects for dealing in terms of broad tendencies and magnitudes ("this strategy appears to increase the thermal efficiency significantly") because many of the choices they must make are for issues which are qualitative rather than quantitative. Thus, different members of the design team may wish to view performance information differently.

A 1991 Construction Industry Computing Association (CICA) study [Howard 1994] into the "extent of uptake of models, perceived benefits and ways forward for developers" recommended that professional institutions become pro-active and endorse modelling and give guidance on use and selection. Since then, CIBSE has followed such a course of action.

In comparison with direct interaction with developers and tool users, published surveys of user demands and opinions are constrained in their information content. Extended interviews with the author of a particular survey can clarify its findings but such data should be balanced by direct observation and interaction. In the context of the current work, direct contact has been achieved in two workshops organised jointly by the U.S. Department of Energy and the U.S. Department of Defence [Crawley and Laurie 1997, Crawley et al. 1997]. These workshops set out to advise on the next generation features that tool developers and users considered were important. Observations of the first workshop, and long interviews with participants during and after the workshops [Crawley and Lawrie 1997] have identified a number of relevant issues.

The first workshop in August 1995 was charged to reach a consensus on the application domains, capabilities and methods for simulation from the viewpoint of the tool developer and expert. The principal goals identified were:

- to facilitate collaborative and integrated building design;
- to educate students and practitioners;
- to evaluate comfort, check code compliance and assess environmental impact;
- to enhance tool capabilities in terms of air flow modelling, flexible systems-side modelling, moisture tracking, multi-dimensional transient conduction and daylighting;
- to adapt interfaces to user type and the stage of the design process, and to provide intelligent defaults, libraries of components and object oriented representations.

The second workshop in June 1996 was attended by simulation users and focused on applications for tools, capabilities and interfaces. This workshop indicated demands for:

- envelope design, early analysis of design alternatives and environmental impact assessments;
- evaluation of contractors who manage energy use for building owners;
- economic analysis;
- enhanced tool capabilities in terms of envelope/ environment interaction, better models of heat transfer and air movement;
- interoperability with other tools (especially computer aided design), flexible libraries of components, context help, and the ability to customise inputs and reporting.

Overall the conclusion was drawn [Crawley and Laurie 1997] that "although the expected bias of the two groups [developers and users] is discernible, there is remarkable agreement on program application priorities". Few new ideas were reported and it was observed that developers were reluctant to expand the boundaries of simulation until fundamental issues had been resolved.

Donn [1997] reports on the use of simulation tools within a survey about practitioner perceptions of quality assurance issues and procedures, expertise required and desired improvements. More than two thirds of respondents never calculated comfort indices.

> *"The picture that arises is of a group of consultants who routinely study capital and running costs. They can do more, but they are normally not paid to."*

In addition to the above, it is clear that constraining perceptions have evolved within the community of practitioners:

- simulation is costly and slow,
- simulation requires special and expensive equipment,
- simulation is a specialist's tool, useful only for high value commissions,
- there is a poor match between measurements and predictions.

Each of these statements might be true for a particular combination of design team, project and assessment tool. Even the evolution of computer equipment, which now requires only a modest investment to support simulation based design, does not appear to have altered the perception that simulation is a costly exercise.

The actions of the simulation community may have reinforced perceptions that simulation is ill-suited to the design process. Many empirical tests, analytical studies, surveys of tools and inter-model comparisons [Judkoff et al. 1994, Lomas et al. 1997] have concentrated on abstract single or two zone problems which have nevertheless been time-consuming and expensive. The design professions might well have concluded that simulation is not capable of supporting realistic models.

1.2 Consequences of the use of simplified tools

> *" ... as modelling sophistication diminishes, so many of the active flowpaths are degraded or ignored and the methods becomes indicative, not predictive, application limited, not general, and of low integrity vis-a-vis the real world" [Clarke 1985].*

Based on the coordination of several hundred energy assessments, the UK Energy Design Advisory Service (EDAS) [McElroy 1997] can put forward evidence that practitioners require access to a palette of tools, each fit for a particular stage of the design process or type of design question. The CIBSE Applications Manual — Building Energy and Environmental Modelling [Hand et al. 1998] — lists as necessary skills: selecting assessment tools, determining appropriate assessment metrics, and proper application of the appropriate tools. In line with these recommendations, managers of simulation based projects must be in a position to specify and deploy appropriate simulation tools and staff resources. Yet EDAS staff report that design teams maintain only a limited number of support tools and do not routinely switch assessment tools during a project.

First and second generation (simplified) tools are relevant for the exploration of early design options where descriptive information is limited and identification of trends is required. The market provides any number of tools which design teams may use for such a purpose. As design questions become more specific and the details of facades, environmental systems and controls become important, a different class of tool is required. However, best practice design does not often extend to detailed assessments based on third and fourth generation tools. Much of this has to do with the pragmatic constraints of the design process. Those who purchase design advice tend not to be willing to pay for detailed assessments or to delay the design process to wait for the results of such deliberations. The remainder of this section considers what the unintended consequences of such pragmatism might be.

Consider a subset of the thermophysical interactions as shown in Figure 1.1. When the built environment is subjected to changing boundary conditions, control and use patterns, its thermophysical state evolves from the dynamic flows of energy via convection, conduction and radiation as well as from air flows resulting from temperature or wind-induced pressure variations.

Such interactions are ubiquitous and exist irrespective of the size, geometric complexity or composition of the design. Choices made during the design processes influence these flow paths and thus contribute to some degree to the resulting performance—be it marginal, acceptable or exceptional.

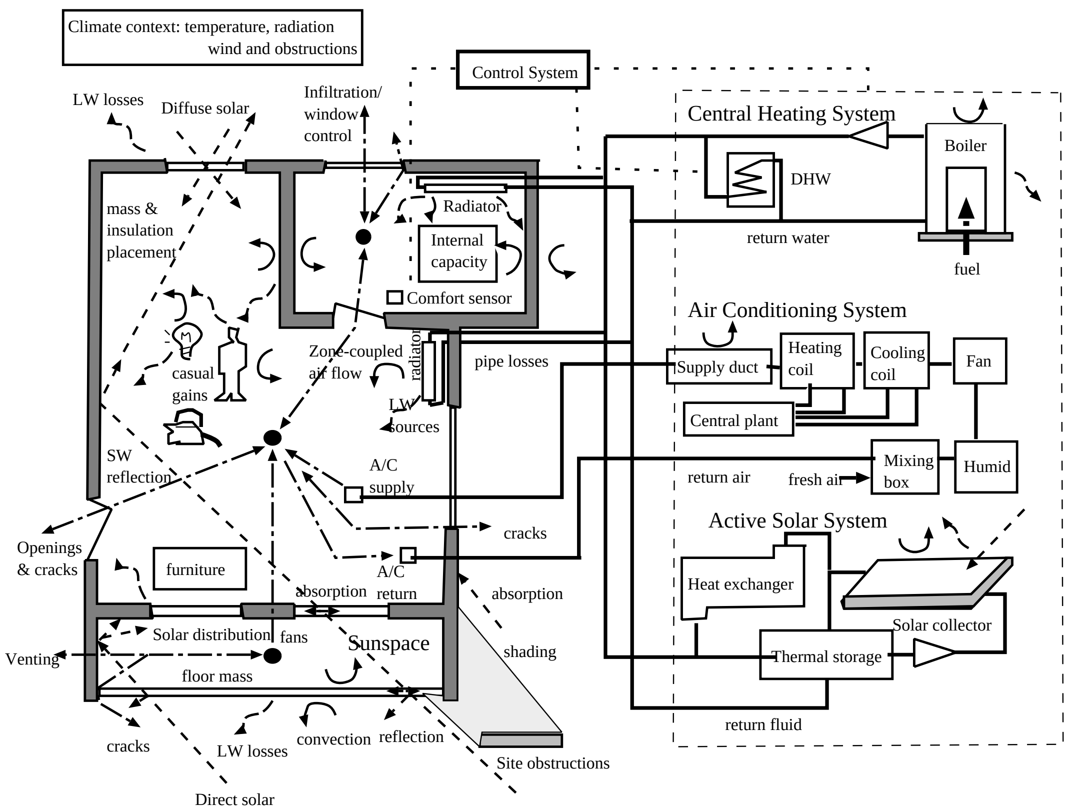


Figure 1.1 Typical flow paths.

Many assessments based on first generation tools assume that in the winter there is no sun, it is always cold outside, the wind never stops blowing and there are no occupants or interactions with adjacent spaces (indeed, only the building envelope takes part in the analysis). Clearly, such assumptions have little to do with the physics of buildings, rather they provide a low risk (conservative) estimation of equipment capacity (which may increase capital and running costs). Many assessments based on second generation tools assume static conditions for natural ventilation—as if the movement of air over a surface would not influence the temperatures of both.

Another tendency in constrained tools is to focus on a subset of thermophysical processes (air flow, conduction, moisture flow, radiation exchange, etc.) in ways which predetermine the importance of physical aspects of a design and the interactions between its elements. An empirical and theoretical study of mass placement [Burch et al. 1985] concluded that "the exclusion of interior mass surfaces from computer predictions causes the benefit of energy conserving modifications in houses to be underestimated." Three variants of this are common.

- In some building assessments, the influence of the plant system is more or less neglected by over-simplification; it is common practice to base the estimation of energy consumption on some imposed indoor air temperature profile. Most building-side tools require, as input, infiltration rates and the flow of air between rooms—parameters that are not easily established.
- In some plant assessments, complex energy flow paths are grossly simplified and regarded as a simple plant load imposition.
- In some air flow assessments, boundary conditions are fixed and the analysis is focused on a moment of time. This precludes the study of potentially important interactions between the air flow, the building, controls and plant systems.

Even where assessments combine several domains, the loads, systems and flows are often treated sequentially rather than simultaneously. Without information on the *interaction* between the building and system, performance optimisation inevitably focuses on systems rather than on the design as a whole. This may lead to a conservative design or a design which does not respond well to control actions.

A survey [Beranek and Lawrie 1989] found that tools in 1989 made use of technology from the previous decade and tended to impose use patterns (e.g. batch mode operation with limited output facilities) which were not in keeping with practitioner demands. One of the arguments for "starting from scratch" was the persistence of such code and the difficulties which this imposed on the maintenance, evolution and use of tools [Crawley 1998].

Thus, there are specific limitations in first and second generation tools which limit their applicability and constrain the information available to the design team.

1.3 Applicability of simulation

It has been argued by Clarke [1994] that simulation now defines a best practice approach to design. Institutions such as the CIBSE, who have long championed the use of traditional assessment techniques, are beginning to recognise the applicability of computer based assessments and wish to guide their members in the selection and use of such tools.

The case for the inclusion of simulation within the design process is supported because:

(a) designers are exploring new design concepts and relationships which are not within the scope of simplified methods and correlations;

(b) design questions often involve a sequence of specific topics which imply a level of descriptive detail, metrics of performance and viewpoints into performance data which cannot be accomplished within the constrained environment of first and second generation tools;

(c) the substantial contributions to CO_2 emissions related to the built environment require analysis support in order to ensure that government targets for the control of gaseous emissions and demand-side energy management are achieved.

The question is—can simulation keep up with the design process, without requiring onerous concessions from the design teams and without compromising the quality of the simulations? Many vendors of simulation tools claim that their programs offer substantial functionality in support of the design process. They have invested heavily in making their applications easy to use and able to address issues of greater complexity.

A related question is whether the advent of commercial interfaces on powerful inexpensive computers has actually removed the mystique and complexity of simulation [Parand 1996]. For this to be true one would have to accept that the translation of a design into a simulation model requires few interpretative skills, that simulation tasks are essentially mechanistic, and that demands of the design process can be reasonably anticipated and automated.

An initial conjecture is that such statements are not true and that ease-of-use has not altered the nature of the expertise required to undertake design decision support or the essentially niche activity that simulation represents to the design professions.

Clearly, the design professions and tool vendors have not yet come to a joint understanding of what is required to support the design process and there exist barriers to effective computational support of the design process.

It is the task of the current work to explore aspects of design which have not been well served by existing tools, training and simulation practice, as well as the nature of the barriers to the routine use of simulation within the design process. From such observations, alternative approaches have been tested and a simulation environment which is more in keeping with the needs of the design process has been introduced.

1.4 The method of the study

Academic developers of simulation tools tend to write about numerical techniques, new environment system components or about how object oriented models and solution methods presage a new era in simulation tools. It is less usual to find a paper that addresses simulation use within the design process or how simulationists can be given the skills needed to contribute to the design process.

The present work is based on a series of case studies, which begin as platforms for the observation of simulation use within design, progress through projects which allow alternative techniques and facilities to be tested, and culminate in projects which allow an evolved simulation environment to act as a virtual laboratory addressing the needs of advanced design projects. The sequence of case studies, each with specific objectives, allows the discussion to traverse issues relevant to simulation use within the design process.

In most of these case studies, the inclusion of simulation in the design process was driven by economic, productivity or comfort issues and this allowed conjecture and testing at a scale and with users typical of the audience whose needs simulation must address.

Chapter 2 begins with a review of the simulation environment, its facilities and data model, and a description of issues related to its use. The case studies in Chapter 3 seek to identify what it is that constrains the use of simulation within the design process. The case studies also are used to enquire whether common perceptions that simulation is restricted to prestige projects, and is an inappropriate and costly tool within the design process, have a basis in reality and how this is manifest in design projects. In particular, these case studies explore:

- how simulation approaches supported or failed to support design goals;
- the appropriateness of the models generated to the needs of the project;
- how work practices expedited or delayed project deliverables;
- what aspects of the design of the tool were ill-suited to project tasks.

The second phase of the work, which begins in Chapter 4, uses these observations in the definition of a simulation environment which better serves the needs of the design process. Chapter 5 introduces the notion of *project management* to simulation and describes how this has been implemented and the extent to which this allows simulation to address problems of realistic complexity within the design process. This is followed, in Chapter 6, by a discussion of how the integrated performance of designs can be supported and understood and lastly, Chapter 7 discusses the knowledge based integration of performance assessment within the modelling process, and lastly, Chapter 8 returns to issues of skills acquisition.

References

Beranek D., Lawrie L., "Promising (And Not so Promising) Developments in Energy Analysis Software" *Proceedings IBPSA BS ’89 pp5-10, Vancouver, BC August 1989.*

Burch D.M., Walton G.N., Cavanaugh K., Licitra B.A. "Effect of Interior Mass Surfaces on the Space Heating and Cooling Loads of a Single-family Residence", *Thermal Performance of the Exterior Envelopes of Buildings III*, ASHRAE/DOE/BTECC, pp239-254, Clearwater Beach, Florida, December 2-5 1985.

Clarke J. A. "The ESP System: Towards a New Generation of Building Energy Analysis Program", *Proc. First International Building Energy Simulation Conference* Seattle, pp215-227, August 21-22 1985.

Clarke J.A. *The Future of Building Energy Modelling in the UK*, A report to the Building Sub-Committee of SERC, 1987.

Clarke J.A., Maver T.W. "Advanced Design Tools for Energy Conscious Building Design: Development and Dissemination", *Building and Environment*, Vol. 26, no. 1, pp25-24, 1991.

Clarke J.A., Private communication, Glasgow 1994.

Crawley D., Lawrie L. "Workshops on Next-Generation Building Energy Simulation Tools: Part 2: Constrasting Developers and Users", *IBPSA News*, Vol 9 No 1, April 1997.

Crawley D., Lawrie L., Winkelmann W., Pedersen C. "What's Next for Building Energy Simulation-A Glimpse of the Future"Proc. BS '97, Prague, September 8-10 1997.

Crawley D., Private communication, Washington D.C. 1998.

Department of Energy, Web page: <http://www.eren.doe.gov/buildings/tools_directory/>

Donn M.R., "A Survey of Users of Thermal Simulation Programs" Proc. BS '97, Prague, Vol III pp65-72, September 8-10 1997.

Hand J.W., Irving S.J., Lomas K.J., McElroy L.B., Parand F., Robinson D., Strachan P. *CIBSE Application Manual AM11 Building Energy and Environmental Modelling* (London: Chartered Institution of Building Services Engineers) 1998.

Howard R., Wager G. and Winterkorn E. "Guidance on Selecting Energy Programs" (Cambridge: Construction industry Computing Association) 1994.

Howrie, J. "Building Modelling: An Architect's View" *BEPAC Newsletter*, Building Environmental Performance Analysis Club, No. 12, pp8-10, Spring 1995.

Johnsen K., Grau K, *TSBI3 Computer Program for Thermal Simulation in Buildings User's Guide (Version B08)*, SBI, Danish Building Research Institute, 1994.

Judkoff R., Neymark J., Van de Perre R., et al. "A Testing and Diagnostic Procedure for Building Energy Simulation Programs" *Proceedings of BEP '94* pp13-116, Reading: Building environmental Performance Analysis Club, 1994.

Kusuda T. "NBSLD, the Computer Program for Heating and Cooling Loads in Buildings", *NBS Building Science Series 69*, July 1976.

Lomas K.J., Eppel H., Martin C.J. and Bloomfield D.P., "Empirical Validation of Building Energy Simulation Programs" *Energy and Buildings* 26(3), pp253-267, 1997.

McElroy L., "The Energy Design Advisory Service as an Aid Toward a New Working Frame", *Proc. 3rd European Conference on Architecture*, Florence, May 17-21 1993.

McElroy L.B. Private communication, Glasgow, 1997.

Parand F. Private communication, BRE, Garston 1996.

Robinson D. "Energy Model Usage in Building Design: a Qualitative Assessment" *Building Services Engineering Research and Technology* Vol. 17 N pp89-95, 1996.

Selkowitz S., Haberl J., Claridge D., "Future Directions: Building Technologies and Design Tools",*Proceedings ACEEE Summer Study 1992*. Vol 1, pp269-290, 1992.

2 ESP-r: A REVIEW

"To see where we might be going, let us look at where we have come from." J.A. Clarke, The Future of Building Energy Modelling in the UK, 1989

The tool selected to provide a platform on which to undertake the work is ESP-r. As shown in Figure 2.1, this system comprises a suite of applications which provide facilities for the description of simulation models, and for the management of databases, simulations and results analysis.

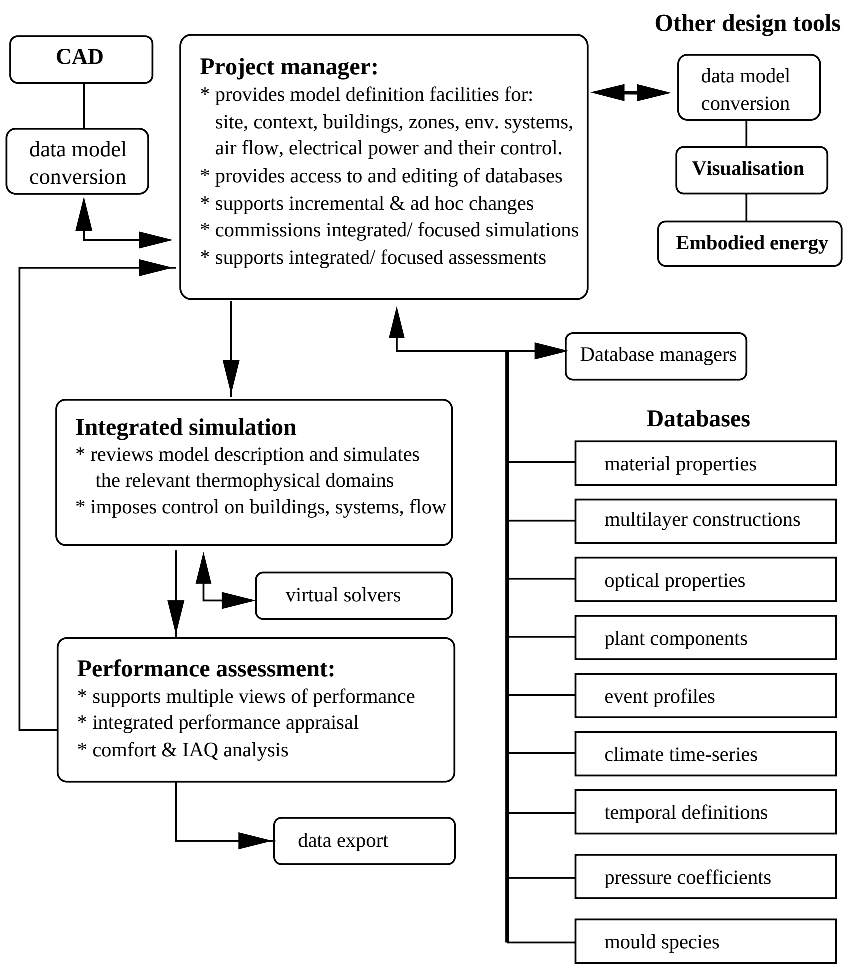


Figure 2.1 The ESP-r suite.

ESP-r is a comprehensive simulation environment which can assess problems related to several domains, namely thermal, air and moisture transport within physical spaces (typically buildings), fluid flow within HVAC systems, electrical power flow within heterogeneous networks (i.e grid and renewable energy connected), as well as environmental control systems and indoor air quality (IAQ) issues (e.g. comfort and mycotoxins). These domains can be assessed jointly or severally as required.

Within ESP-r, it is also possible to select approaches to domain solution—one, two or three dimensional conduction; a mix of scheduled air flow, network or computational fluid dynamics (CFD) for flow assessments; a mix of ideal or explicit representations of plant and control systems. Within a single model the user may define one or more levels of geometric detail from, say, a thermostat to a cluster of buildings. The temporal resolution of the solution of each domain can also be specified from fractions of a second to an hour, and varied during the assessment (even to the point of backing up to the start of the day to explore an alternative control strategy).

In order to illuminate the issues raised in the case studies as well as the nature of the conjectures and tests undertaken, the discussion begins with a review of ESP-r and its data model. The scope and functionality of modules in the suite is described in Table 2.1. The rationale for distributing functionality is derived from the nature of the tasks to be performed—maintaining databases, describing the problem, proving the model, extending its robustness, commissioning simulations and understanding performance predictions.

Table 2.1 System modules.

Module	Description
problem creation	provides facilities to instantiate and check the syntax of the simulation model in terms of the site, context, networks and control as well as zone details and documentation
model viewer	provides hidden line and coloured/ textured views of the problem geometry for visualisations and shading studies
database managers	provide facilities to manage and manipulate database entities (elemental thermophysical properties, composite constructions, plant components, typical profiles, etc.)
climate	provides facilities to view, analyse and manipulate climate data sets
domain preprocessing	computes temporal shading and insolation patterns and view factors between surfaces in zones (to support more rigorous assessments)
integrated simulator	given the current problem, context, databases—solves the thermal/air flow/power domains, imposes control as required and generates a database which holds the results for each simulation time step
performance assessment	supports the exploration of performance via graphic and tabular presentations of state variables as well as derived indices for comfort and overall energy demands

At the core of the integrated simulator are numerical solvers optimised for the building, network or CFD based flow, plant and electrical power domains, considered separately or in various combinations, at time steps ranging from seconds to an hour. Each of the solvers is linked by message passing conventions so that, for example, the results of the iterative network flow solution are made available to the building solver and vice versa.

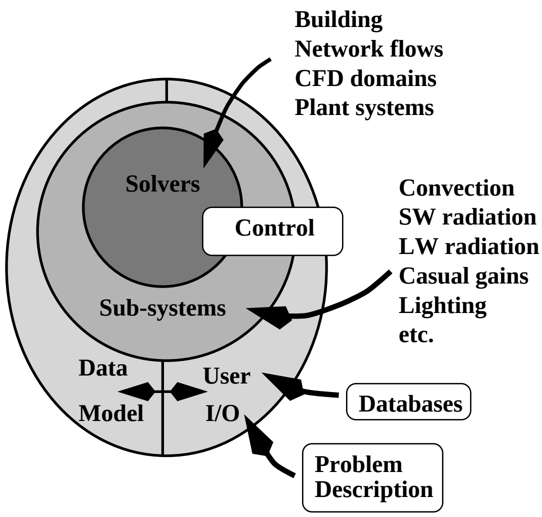


Figure 2.2 The integrated simulator.

As seen in Figure 2.2, the core solvers are surrounded by the subsystem code layer which deals with solar radiation, convection, etc. Outwith this is a layer which converts the description of the current model, databases and control specification into the data model used by the solvers. The outermost layer supports interaction with the user. Note that control bridges each of the layers because it is dependent on information at each level and usually needs to coerce entities in the data model, subsystems or solvers.

A definitive description of ESP-r [Clarke 1985a] and the essential difference between its solution technique and that of response functions and frequency domain approaches [Clarke 1985b] has been expanded by contributions from a number of sources [Hensen 1991, Aasem et al. 1992, Aasem 1993, Chow 1995, Nakhi 1995, MacQueen 1997]. The current bibliography with a more extensive list of references to ESP-r's treatment of network air flows, computational fluid dynamics, dynamic plant systems, controls, electrical power and other domain theories is held in hypertext form [ESRU 1998]. The representation and numerical solution of physical processes has been extensively covered elsewhere, and such topics will be mentioned only in so far as they promote or hinder the use of simulation tools as addressed in the current work.

2.1 The data model

One of the central tasks in undertaking an assessment with any tool is the representation of the essential character of the design in terms which are understandable both to the user and to the tool. Thus, for example, a conference room becomes a collection of data model entities which must be both syntactically and semantically correct in terms of their own detail and in relation to other entities (e.g. rooms, environmental systems, site details, etc.).

The mapping of design details (form, composition, operation) into data model entities varies from tool to tool. For example, a simplified tool may constrain the model to two zones, each rectangular and with a single type of glazing system and composition for all external walls. It may also offer several "pre-cooked" environmental control systems, which are particularly appreciated in early-stage studies. In such a regime there is a specific mapping of choices offered to the user into the internal data representation.

In simulation in general, and ESP-r in particular, few aspects of either the design or the simulation model are pre-determined. The user thus has the freedom to define the specific aspects of the design which are to be considered within an assessment from the entities shown in Figure 2.3. Where more information is required, simulation allows the user to increase temporal, compositional or geometric resolution.

In ESP-r, a simulation model comprises a context with one or more buildings, each of which may be composed of one or more thermal zones which have geometric, operational and constructional attributes. There may be one or more technical networks and control regimes within a model. Thermophysical meaning is derived from both attribution (e.g. a wall has a name, composition, surface properties, orientation) and contiguity (e.g. a wall joins other surfaces and has boundary conditions at each face). To aid understanding, the data model includes a number of entity naming and documentation conventions.

As will become clear in the case studies, the user’s freedom to draw from this store of entities requires some understanding of the data model in order to ensure that the simulation model is an appropriate abstraction of the design and is able to provide support for design questions. For this reason it is worth reviewing the components of the data model shown in Figure 2.3 in some detail.

Site: The site comprises information related to position and the surrounding land forms and the context within which assessments take place.

location: includes the latitude and longitude difference (from the nearest time zone reference meridian) as well as the exposure of the site (used to define external view factors for long-wave radiation exchange). If the topology of the site is considered important, land forms can be explicitly represented.

context: is the repository for documentation and images which clarify the nature of the model as well as the patterns of boundary conditions—outside dry bulb temperature, humidity, wind speed and direction (and the resulting distribution of pressure around buildings) as well as direct and diffuse solar radiation. The model context may also hold information on fuel sources, tariffs and applicable power generation fuel mix and emissions data.

Building: One or more buildings may be included in a model. There is no demand that all zones are contiguous, neither is there a demand for buildings to be represented at the same level of resolution.

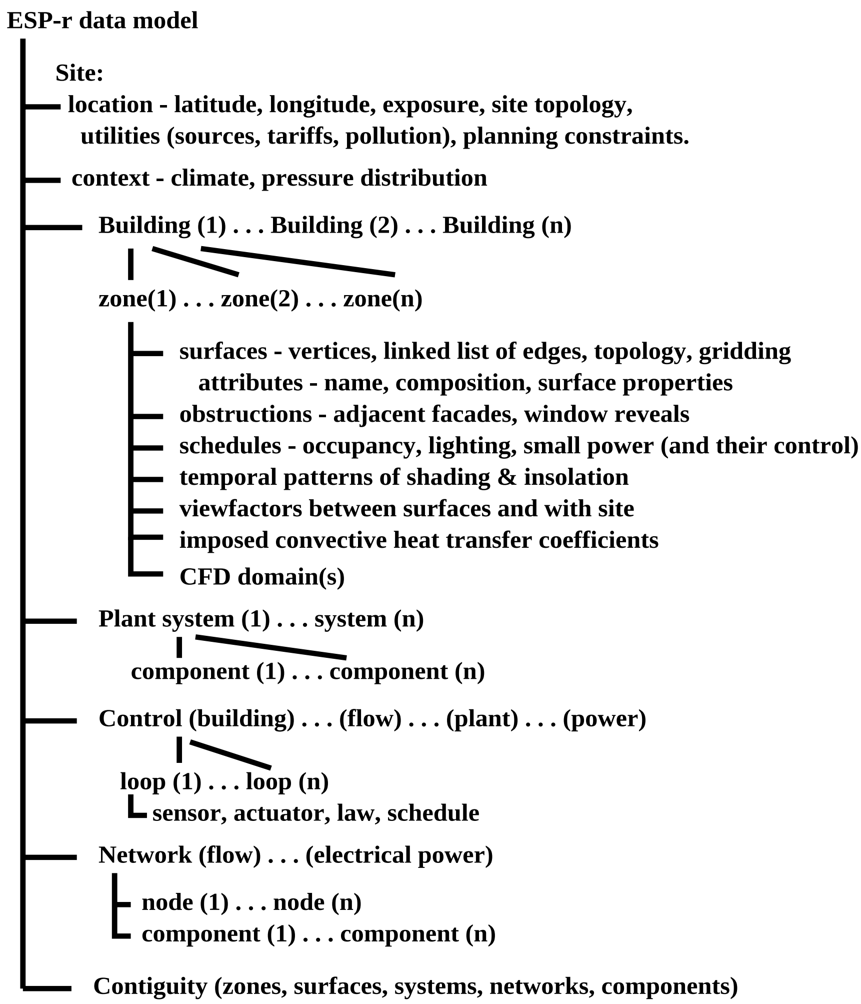


Figure 2.3 ESP-r data model (synopsis).

thermal zone: is a fully bounded body of air (or other fluid) at a uniform temperature which has attributes of form, composition and operation. Each zone is composed of a polygon enclosure, into which additional surfaces may be placed to represent internal mass. A zone might comprise the enclosure around a thermostat, a room, a floor of a building or the building itself.

surface: is a plane polygon of one construction, with one face associated with a zone and the other face related to one boundary condition (external, another zone, a similar but hypothetical zone, a constant or monthly temperature profile or an adiabatic state). Each polygon is defined as an ordered list of vertices (the ordering determines the orientation), and each edge must be shared with another surface. Just as a zone may represent spaces at different scales, a surface can represent all or part of a wall, frame, furniture or fenestration.

obstructions: are rectangular bodies used singularly or in combination to represent objects which may obscure solar radiation, but which do not otherwise participate in thermal transfers. Obstructions may also be used to enhance the visual resolution of a model.

operations: are schedules of internal gains from occupants, small power and lighting associated with a zone. Gains are expressed in terms of one or more periods during weekdays and weekends (typically, but not necessarily, Saturday and Sunday) and each includes the magnitude of sensible and latent gains. Sensible gains are further specified by their radiant and convective split. Schedules can also be used to specify patterns of infiltration and ventilation as air changes per hour. Such schedules can have imposed control based on one of several boundary or internal condition tests.

shading and insolation: are temporal patterns of shading on external surfaces caused by obstructions and insolation entering and falling on surfaces within a zone. Tracking the temporal evolution of these patterns is computationally intensive and so they are held in a database and imposed as the assessment progresses.

view factors: are the explicit radiation exchange factors between surfaces within a zone which support a robust treatment of longwave radiation exchange and detailed comfort studies. Because zone enclosures are explicitly defined it is possible to pre-calculate and hold these factors as a matrix.

convective heat transfer coefficients: define the rate of heat flux exchange between a surface and the air. They are normally evaluated at each time step, but can be imposed on specific surfaces if the user has prior knowledge (measurements) or wishes to test sensitivity.

CFD domains: define gridding schemes and computational parameters for static or transient computational fluid dynamic domains within a model. Such assessments are an option where detailed air flow or heat transfer mechanisms are of interest.

Plant systems: Where detailed treatment of environmental systems is of interest, explicit plant models can be described as a network of components which are linked to zones.

plant network: is a set of components linked by user defined connections. Flow in a network can be explicitly determined by mapping each connection to a network flow connection to be resolved by the network flow solver.

component: is a finite volume representation (instance) of a plant entity based on a template from the plant component database.

component subroutine: code which takes the plant component template and user supplied data and generates the matrix equation for the solver.

component containment: is the environment surrounding the plant component (e.g. another component, a zone, or outside) to which heat is lost.

plant connections: define the paths for working fluids and flux. Each path can convey thermal energy, water, dry air or moisture as determined by the components at each end of the connection.

Network flow: Where the design requirement is to assess flows which result from changing boundary conditions two methods are offered—network (bulk) flow and the previously described CFD domain.

Many design questions related to natural and forced ventilation can be addressed by taking into account bulk flows of air, and the leakage paths and boundary pressures and buoyancy which influence flow within a building. Network flow is defined via boundary and internal nodes linked by flow components (pipes, ducts, fans, openings).

node: represents a volume of air or liquid which is internal (with a known pressure or a pressure to be solved) or a boundary condition (with a known pressure or wind induced pressure). Nodes are given a specific temperature or can inherit the temperature of a zone as a simulation proceeds.

component: is a flow inducer (fan, pump, etc.), conduit (pipe, duct, etc.), restrictor (orifice, crack, valve, etc.), or diverter (junction). Currently there are 22 component types, derived from empirical tests or analytic methods.

network of nodes and components: is defined by a list of connections of the form: "*office (node) is connected to passage (node) by way of door (component)*" such that a path exists from each node to at least one boundary condition.

Network power: The flow of power may also be defined and solved with a network analogy similar to that of air flow, but with busbars replacing pressure points, conductors and transformers replacing flow components and loads and sources defining boundaries.

components: are loads (small power, equipment, lighting) and generators (combined heat and power, PV modules, etc.)

casual gains: such as lighting and small power comprise both a thermal and an electrical definition (voltage, real and reactive power, power factor, phase).

special materials: are surfaces which have additional attributes and behaviours. For example, power sources such as photovoltaic modules are affected by incident radiation and temperature (derived from their thermophysical and radiant context). However, when the context is a building facade, they tend to modify the thermal and radiant characteristics of the building. Detailed assessments thus require electrical, thermophysical and optical properties to be taken into account and for the building, flow and power solvers to work cooperatively.

busbars: define the power balance points between loads and generators.

Control: As noted in Figure 2.2, control can be imposed on several aspects of a model—zones, flow, plant and power. The highly coupled nature of building physics requires that simulation solvers be subservient to control logic and that control logic allow considerable flexibility in operational detail.

building control is sometimes referred to as *ideal* control and is defined in terms of sensors (at a boundary, zone air node, at or within a surface) and actuators (convective to zone air node, radiant/ convective mix to air node and surfaces, at or within a surface). For each sensor/ actuator combination, there is imposed a schedule with each period indicating a control law (e.g. ideal, optimal start, multi-stage, fuzzy logic) and operational details (set points, capacities, etc.).

This syntax supports simple assessments (e.g. how much convective heating is required to maintain 20°C between 8h00 and 18h00 on weekdays), fuzzy logic schemes which only control engineers would be in a position to specify, or lighting switching control based on combined lighting and thermal assessments.

flow network control, which adjust to changing boundary conditions and approximate use in practice (e.g. occupants opening windows), can be imposed on particular flow components, allowing them to be switched in on/ off mode or incrementally adjusted. Scheduling is similar to that of zone control.

plant modelling control supports detailed assessments of equipment use and control system design (including high-frequency feedback loops within environmental systems).

Contiguity: The generality of simulation and the flexibility of model definition is resolved by an extensive and strict definition of contiguity within a model (e.g. surface edges in zone enclosures must adhere to topological conventions, boundary conditions must be specified and be consistent at each side of partitions, flow networks may not mix fluid types, and changes in databases must be applied to all referencing entities). This happens at the descriptive levels of buildings, zones, surfaces, networks and controls.

between buildings: e.g. buildings may shade each other and they may share elements of a flow network (e.g. district heating).

between zones: e.g. a matrix of zones, identical except for a control scheme or wall composition might be composed so that commonality derives from shared descriptive files.

between surfaces: e.g. contiguity may be explicit (ceiling in office 311 is in contact with floor in office 411) or implied (ceiling in office 311 is in contact with another zone with similar conditions). Explicit contiguity can often be derived from geometric adjacency.

between networks: e.g. flow networks in separate buildings may be solved independently or may share flow paths. Flow network nodes may be linked to thermal zones and thus inherit their temperatures and impose infiltration and ventilation loads. Flow networks with dissimilar working fluids may exchange heat via a heat exchanger plant component. Power networks track power production in photovoltaic (PV) modules which convert incident radiation into power and heat.

between controls: e.g. operation of windows in a flow network can affect temperatures in zones and lead to the invocation of a heating regime.

It transpires that ESP-r has one of the most extensive data models of this class of tool [Clarke et al. 1995]. It is also the case that similar entities can be found in other assessment tools, albeit that each vendor/ developer evolves different formats and thermophysical relationships between entities. In the case of ESP-r, the simulation model is held in a distributed file store which closely matches the data model. System level files hold the context of the model, controls, networks and the like, while each thermal zone has files for geometry, thermophysical composition and schedules.

2.2 Databases

Databases (such as those shown in Figure 2.4 and described in detail in Table 2.2), are an adjunct to the data model and represent a common store of information used across a range of projects They are held in a secure location and in a form which limits the possibility of corruption. In the case of project specific information (e.g. measured thermophysical properties for a test cell) or project confidential information, local databases may be created.

ESP-r databases

- **Materials (aluminium, plaster, etc.) thermophysical properties**
- **Composite constructions (walls, glazing systems, floors, etc.) - referencing materials and optical properties**
- **Optical properties - visual and solar transmission and absorption**
- **Climate sets - hourly data on temperatures, radiation, wind, humidity**
- **Pressure sets - pressure distributions (by angle and exposure type)**
- **Events - profiles (representing casual gains and controls/regimes)**
- **Temporal shading / insolation - hourly data (typical for each month) per zone**
- **Plant components - (boilers, pumps, heat exchangers, etc.)**
- **Moulds - (data defining growth limits)**

Figure 2.4 ESP-r databases.

Some databases, such as the materials and climate databases are self-contained, while others, such as the composite constructions database, are dependent on other databases. Because the number of entities may be large and access is required both during the creation of a simulation model and at intervals during the simulation process, databases are random access.

Table 2.2 Database entities.

Database	Contents
materials	Comprises a number of classifications (e.g. wood, concrete, metal) each of which holds a set of materials and their thermophysical properties (density, specific heat, conductivity, moisture permeability, shortwave absorption and longwave emissivity).
multilayer constructions	Named constructions (walls, floors, glazing systems, etc.) held as ordered lists (from outside to inside) of primitive elements and thicknesses. Each construction has an optical properties attribute which, if non-opaque, points to a named entity in the optical properties database.

Database	Contents
optical properties	Named sets of optical properties (visual transmission, solar band transmission at 5 angles and angular absorption at each layer) which is referenced by the multilayer construction database as well as blind control definition and actuation facilities.
mould growth	Defines growth profiles (temperature and humidity) for species of mould.
pressure distributions	Pressure distributions for wind at 22.5° intervals for surfaces of particular orientations and aspect ratios (width to height) usually derived from empirical tests.
plant components	Plant components which are defined in terms of each component's name and description, finite volume definition (sub-matrix set-up and coefficients)) number and type of connections and their working fluids, thermophysical and flow data and control parameters which are accessed during the definition of the plant network and subsequent simulation.
climate sets	Hourly values of dry bulb temperature, relative humidity, wind speed and direction and either global horizontal and diffuse or direct normal and diffuse solar radiation with a header containing the measurement location identifiers which are accessed by the climate analysis module as well as the simulator.
events profiles	Profiles which are defined as a number of discrete periods and two associated fractions within the range 0.0 to 1.0 which may be imported to define the sensible and latent gains associated with occupancy and equipment.
temporal database	Time step tabular or diary data such as radiation, temperature, velocity and orientation which can be associated with a simulation.

One of the initial and ongoing tasks in simulation work is the instantiation and maintenance of databases. The availability of appropriate data, along with documentation on its source and variability, is for many firms, one of the foundations on which they build successful practices. There are dependencies between databases which must be resolved as new data becomes available or new entities are added. For example, a new glazing system may require a new material (glass with a lower surface reflectance) in the materials database, two new multilayer constructions for the glazing and the frame, and a set of optical properties.

2.3 Data model transforms

Simulation does not exist in a vacuum and facilities to exchange information are an integral part of most simulation environments. The more defined the syntax and clear the semantics of the data model, the higher the probability that a translation function can be written. Translating from a superset data model to a less detailed description is often straightforward. Table 2.3 is a list of the filters which have been implemented in ESP-r.

The filter between ESP-r and a hidden line viewer program (shown in Figure 2.5) is an example of a one way conversion from a superset description (simulation data model) to a hidden line description (polygons) via the native format of the receiving application.

Table 2.3 ESP-r translation facilities.

Conversion	Description
ESP-r > Xzip > ESP-r	converts all geometric entities to and from zip format, ESP-r attribution lost on transfer to Xzip.
ESP-r > DXF > ESP-r	each zone to a separate layer (optional separation of opaque & transparent surfaces), surfaces and ground topology to 3DFACE or 3DPOLY, obstructions to blocks. Reverse translation dependent on user following the above convention.
DXF > Viewer	conversion of most DXF entities to hidden line representation.
XZip > ESP-r > Viewer	two step conversion to hidden line representation.
ESP-r > Viewer	all surfaces & obstructions to viewer bodies for hidden line views.
ESP-r > TSBI3	converts construction databases, zones, surfaces, windows to TSBI3 format.
ESP-r > XFig > Postscript	converts wire frame views to vector commands and Postscript.
Window 4 > ESP-r	imports optical properties.

Somewhat more complex are exchanges between simulation and CAD applications—where the number of entities can be large and where the exchange is bi-directional. Where both tools have similar data models (as is the case of ESP-r and Xzip [Abacus 1991]) the conversion is straightforward albeit that there is still scope for data loss in the conversion (Xzip files do not include topological or compositional attributes).

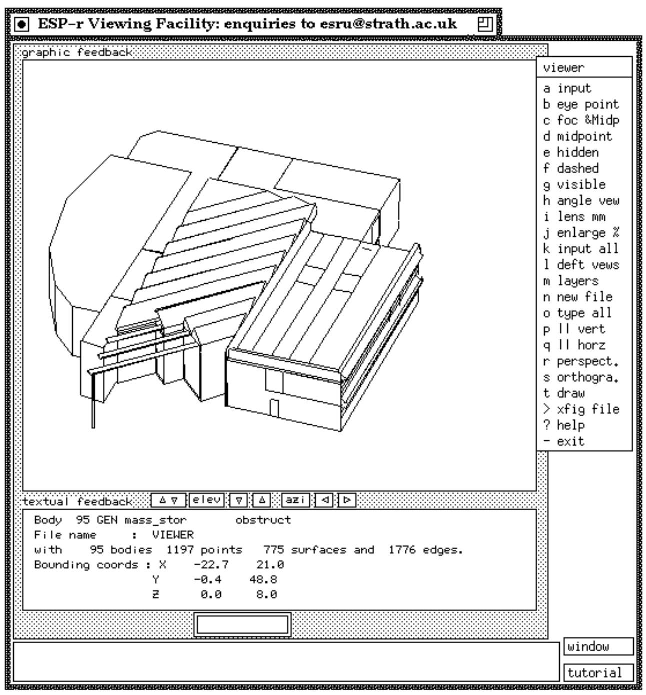


Figure 2.5 Simple transform to line perspective.

Such filters enable aspects of a design to be passed between a limited number of applications with little or no intervention by the user. Where the number of tools is constrained and mature, filters will have some degree of success. As the number of tools increases, the problem quickly becomes intractable. More important, in terms of this thesis, are mismatches at the level of fundamental concepts and the lack of a means to ensure that the semantics of entities are robustly maintained. Without access to both tools' source code, direct mapping of models is inevitably constrained and subject to information loss. A number of approaches to this problem are considered in Chapter 7.

2.4 The descriptive process

Given the depth, breadth, and generality of the data model just described, there are a number of approaches for using such "building blocks" to describe the essential character of a design. Essentially, the user is charged with defining reality within the simulation domain (the "objects" to be simulated, the boundary conditions to be applied and the control or operational regime to be imposed). This usually requires a mental model of the simulation facilities, the descriptive file structure, database contents and dependencies within models (e.g. when shading needs to be recalculated).

Simulation acts on an *abstract description* of a design, via a syntax that expresses the underlying *data model* of the tool. Abstraction is the *process* of arriving at such descriptions—ideally taking into account the methods and goals of the project.

Practitioners who are well versed in the thermophysical nature of the design are in a position to select which parts of the design must be modelled in detail, which may be ignored and which may be represented abstractly in order to answer a particular design question. Those who are less well versed are at a distinct disadvantage; and Chapter 8 explores this issue in detail.

Methodical use of simulation thus takes into account flows of information, decision points, relationships between simulation facilities, and the generation and interpretation of predictions so that the practitioner is in control of the simulation process and focused on the essence of a problem. Indeed, undertaking a simulation based project could be likened to a strategic plan in the form:

> *In order to answer a design query about a building it is necessary to describe the essence of its form and composition, determine assumptions about its use (e.g. occupancy), create simulation models of the base case and design variants in question. Simulations will then be invoked for representative periods in order to explore the differences in performance using metrics which quantify comfort or system capacity.*

Firms or research groups have often derived an overall methodology for approaching simulation which they then adapt as required. Corcoran [1997] defined a six step approach for issues of energy demand and supply in UK hospitals [BDP 1985]. These are:

1. Begin by understanding key functional and comfort related factors that influence energy use.
2. Establish metrics for energy savings and a reference design against which energy savings can be evaluated.

3. Appraise measures to reduce demands for environmental and process systems at the point of use.
4. Determine grades of energy (low/ medium/ high) required for reduced demands.
5. Appraise use of heat recovery to offset energy demands.
6. Appraise measures for meeting residual energy demands and for integration of systems supply.

Clarke [1989] defined a methodology and some of the steps, issues and decision points are included in Table 2.4.

Table 2.4 Steps and decision points.

Step	**Typical decisions**
a: Identify issues to be addressed, simulation and reporting facilities required and indices used to judge performance via consultations between the client, design and simulation teams.	Is the project a one-off assessment? Is it a parametric study or an interactive exploration? Are patterns of external shading and internal shortwave radiation distribution sufficiently described by approximations or should they be dynamically analysed?
b: Abstract the essence of the design into the syntax of the tool, at a level of detail appropriate to the focus of the study.	Is it necessary to describe the whole design or can a portion be studied and the results scaled? How much geometric detail is required in order to capture variations in daylighting which may influence artificial lighting demand?
c: Organise problem files and documentation and proceed with the simulations.	Which databases are appropriate and what additions need to be made for this project? Are there regular patterns of occupancy and equipment use? What naming convention is appropriate if there are three design variations?
d: Prove and calibrate the model so that all parties have confidence in it and the energy signature[1] of the design which emerges.	The computer room temperatures are higher than expected - are the specifications for the equipment appropriate? The rate of cooling lags measurements by six minutes - are the representation and assumptions about flow patterns at odds with conditions within the test?
e: After the simulation, results must be interpreted, performance assessed, reports written and the client advised.	There is an unexpected late afternoon energy pulse in one zone - how may its causes be traced and, when found, how could it be brought before the design team? Can the tool’s native reporting facilities be used or should results be passed to an external package for statistical analysis?

To illustrate the descriptive process—and show how design questions lead to specific assessment tasks and the creation of simulation models–consider the open-plan mid-floor office space shown in Figures 2.6 and 2.7 and the thermal and visual assessments needed to mitigate occupant complaints of poor thermal comfort and glare as well as encouraging better use of artificial lighting.

[1] An energy signature is a pattern of temperatures or flux over time (typically a 24 hour profile) which is in some way characteristic. For example, a school situated in a cold climate might exhibit a high morning heating demand.

Figure 2.6 Open plan office—view from office to core (left) and to atrium (right).

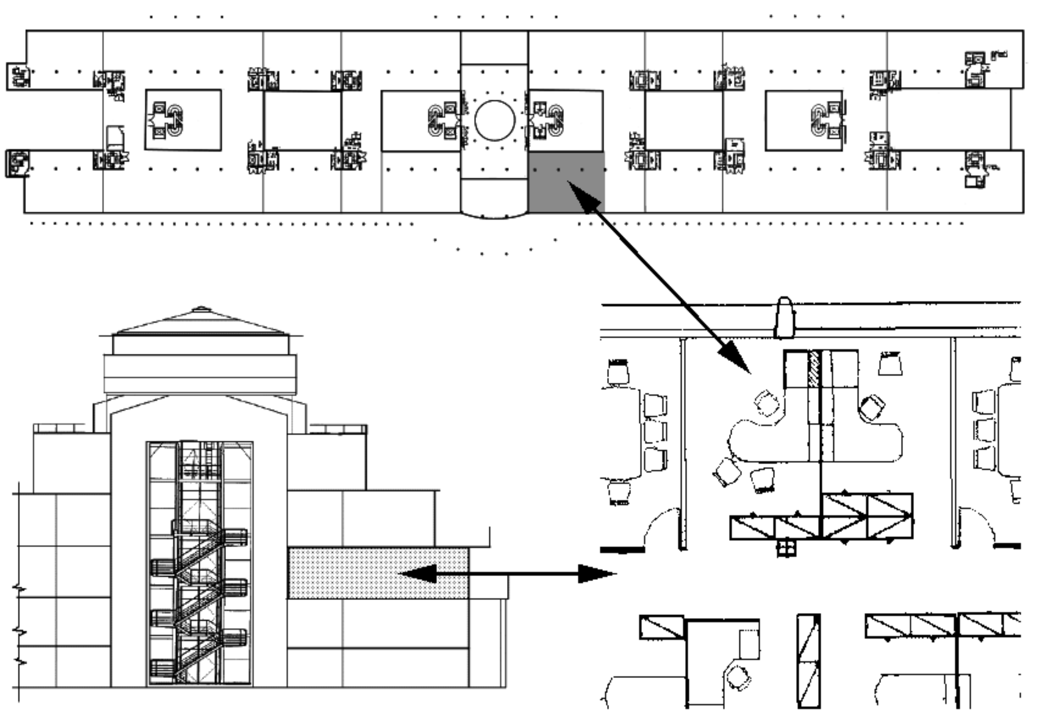


Figure 2.7 Office plan, section, and details.

A number of aspects of the design are important when considering how simulation can be used to answer such questions.

- The north facade of the office is a tempered atrium and it is adjacent to a document processing and storage department which is characterised by clusters of computers and filing cabinets.
- The environmental control system is based on a combination of displacement ventilation, perimeter heating and manually operated windows (there is some cross ventilation potential but the plan depth is 14 m). Structural mass is exposed on most exterior walls and at the ceiling level and within the sub-floor air distribution plenum.

- Like most office spaces it includes a considerable degree of clutter, some of which is thermally trivial but which might be important to the visual environment. The document storage is both thermally and visually important.
- Light (and glare) is borrowed from the atrium and supplemented by banks of ceiling mounted fixtures, those closest to the atrium are sensor controlled. The adjacent document processing group has low levels of daylighting at a number of workstations.
- Patterns of occupancy and space use change every few months as project teams evolve (albeit that the same elements of furniture are used).

How might simulation be used to approach such a problem? Ideally, the user should consider the nature of the questions which arise in the design process, the related simulation tasks, the level of detail needed to inform the design process and the metrics used to judge performance predictions. Taking these in turn:

Design questions might arise from the facilities manager's interest in the sensitivity of the building environment to changes in use and occupant loads and in applying this knowledge when changes are being planned.

Assessments are required to confirm the frequency of occurrence of thermal and visual discomfort, to rank-order the likely contributors and to propose alternative designs or operational regimes.

Metrics of performance might be percentage of people dissatisfied (PPD) [Fanger 1967], number of hours over 24°C during a summer season, and glare source predictions and daylight factor contours for visual assessments.

Assessment domains are building fabric, air flow (for the displacement ventilation and operable windows), and lighting.

Rules of abstraction are that the assessment should focus on the office in question at sufficient detail to answer detailed questions and include other portions of the building at lower resolution to establish boundary conditions.

Thus geometric detail and surface properties are important in the office to support visual assessments. Thermal details on the north facade can be approximated (the atrium is tempered). Assessing the level of borrowed light and atrium temperatures also requires less geometric detail. The displacement ventilation implies that the air is not well mixed within the space and the existence of openable windows introduces uncertainty in the rate of ventilation. Heat plumes from equipment and ceiling mounted lighting fixtures will result in elevated temperatures in the upper portion of the room. The massing of furniture and document storage is important in the space for both visual and thermal analysis.

With this, the practitioner is in a position to select an appropriate tool, and to consider the constraints of the data model and simulation facilities and how a model might be abstracted. Given that a thermal zone within ESP-r is *a volume of air at uniform temperature* and that thermal mass must be explicitly (even if abstractly) represented, the scheme shown in Figure 2.8 fits the project criteria.

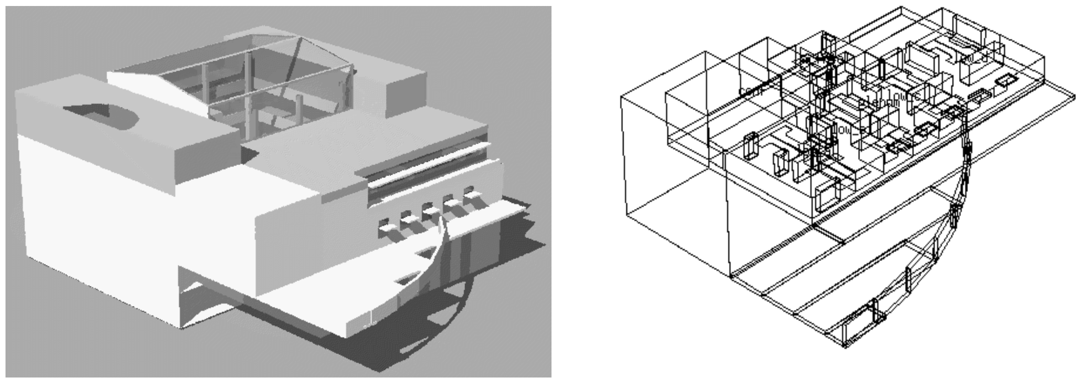


Figure 2.8 Visual model (outer view) and mid-level geometry of thermal model.

The model includes the mass of most desks and document storage in the vicinity of the focus office and beyond that as obstructions within the visual assessment. The office is subdivided into a lower (occupied) zone and an upper zone and the sub-floor plenum is also represented as a zone. The operational regime of the building in terms of small power loads, occupancy patterns and environmental control can be confirmed from design team records and interviews with occupants and facilities managers.

Next, several short period assessments under different operational and climatic patterns are used to test sensitivity and to gain confidence in the model. It is also necessary to observe what assessment time step in each domain yields predictions in keeping with the performance metrics of the project. Particular issues such as surface condensation or air flow through large openings may require short time steps.

The metrics of the project require integrated and frequency-binned comfort and control assessments under a number of regimes. This might be addressed via annual simulations for each parametric combination. However, it is also possible to focus on a subset of operational and seasonal patterns and scale up to annual performance indicators using suitable ratios, e.g. degree days in the case of heating energy.

The visual model, as shown on the left half of Figure 2.9, represents the view from a standing position looking in a direction parallel to the atrium facade with glare sources imposed. The other visual metric, daylight factors, (shown on the right of the Figure 2.9) allows for overall patterns to be observed. The calibration phase of a visual assessment might make use of site survey data to confirm the level of geometric complexity required—for example, facade and atrium roof framing might be deemed necessary while chairs are a needless complexity.

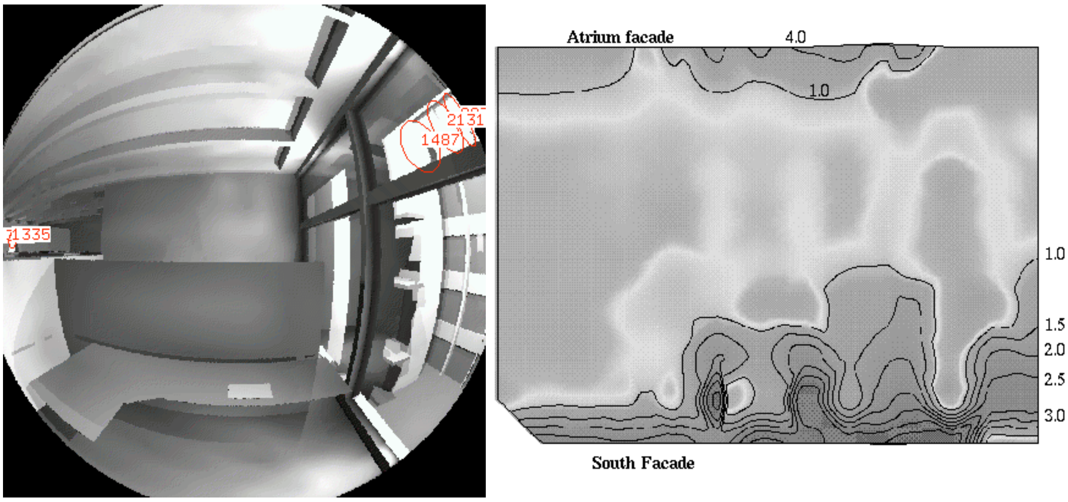


Figure 2.9 Visual model and daylight factor contours.

Having derived a set of performance assessments, the project metrics can be collated, and conclusions drawn and passed to the design team. This may result in proposals for alternative designs and the need to adapt the model and undertake additional assessments. For example, from the visual model, variants can be composed to test the sensitivity of glass transmission, light sensor placement, furniture colour or technology variants such as light shelves. The thermal model might be altered to test different temperatures for the supply air or sensitivity to a given percentage increase in small power loads.

At least, this is what many in the simulation community would hope that simulation could contribute to the design process. Unfortunately, even where projects such as this are within the computational scope of simulation, there are few practitioners with the skills needed to undertake such assessments. The convoluted path by which models are created and the attention to detail required are considerable barriers. Also, requirements for the evolution of models presume much that is simply outwith the current definition of simulation in terms of maintaining model consistency, managing simulation tasks and assessing quality.

If this is true for the design questions faced by today’s practitioners, what then of emerging topics such as the prediction of mould growth or the study of cooperating renewable and conventional electrical power sources? The former, for example, involves the evaluation of local surface temperature/ condensation patterns and the subsequent growth of moulds which are dependent on the physics of vapour transport and absorption within constructional materials. Here, few tools exist for the practitioner and "best practice" has yet to be defined. If current practice is flawed for existing assessment tasks, what paradigm is appropriate for the tools which answer emerging questions?

Indeed, what would be the attributes of simulation-based design decision support that would attract the interest of the design professions as well as those who wish to extend the bounds of design decision support? One task of this work is to identify such attributes and to argue the benefits that might accrue from their application within the design process.

References

Abacus Simulations, *XZIP Xwindows Zone Input Program Reference Manual*, Abacus Simulations Limited, Glasgow 1991.

Aasem E., Clarke J., Hand J., Hensen J., Pernot C., and Strachan P., ESP-r A Building and Plant Energy Simulation System, Version 8 Series, ESRU Manual U92/2, Glasgow, 1992.

Aasem, E. *Simulation of Buildings and Air-conditioning Systems in the Transient Domain.* PhD Thesis, University of Strathclyde, Glasgow, 1993.

BDP, *DHSS Low Energy Hospital Study, Volume 2: Studies and Technical Information, Section G1: An integrated energy approach*, Building Design Partnership, Manchester, 1985.

Chow T.T., *Air-conditioning Plant Component Taxonomy By Primitive Parts*, PhD Thesis, University of Strathclyde, Glasgow, 1995.

Clarke J.A., *Energy Simulation in Building Design*, Adam Hilger Ltd., Bristol and Boston, 1985.

Clarke J. "The ESP System: Towards a New Generation of Building Energy Analysis Program", *Proc. First International Building Energy Simulation Conference* Seattle, pp215-227, August 21-22, 1985.

Clarke J. Private Communication, Glasgow 1989.

Clarke J.A., Hand J.W., Mac Randal D.F., Strachan P.A. *Final Report for the COMBINE II Project: Appendices 4 and 5: The ESP-r Aspect Model*, University of Strathclyde, Glasgow, August 1995.

Corcoran M., Private communication, Glasgow. 1997.

ESRU, Web page: <http://www.strath.ac.uk/Departments/ESRU/>. 1998.

Hensen J.L.M. *On the Thermal Interaction of Building Structure and Heating and Ventilating System.* PhD Thesis, Technische Universiteit, Eindhoven, 1991.

Fanger P.O. "Calculation of Thermal Comfort: Introduction of a Basic comfort Equation" *ASHRAE Transactions*, V 73, part 2, paper 2051, 1967.

Nakhi A.E. *Adaptive construction modelling within whole building dynamic simulation*, PhD Thesis, Department of Mechanical Engineering, University of Strathclyde, Glasgow, October 1995.

MacQueen J. *The modelling and simulation of energy management control systems* PhD Thesis, Department of Mechanical Engineering, University of Strathclyde, Glasgow, 1997.

Chapter 3

CASE STUDIES

3 CASE STUDIES

"The Knot is the heart of all things hidden. Never cut what you can untie."

[Jo Clayton, The Burning Ground, 1995]

Simulation is widely applicable and able to address a range of questions related to conformance to performance based standards, supporting novel designs, best practice and research work (see e.g. Clarke [1993] and Selkowitz [1992]). These areas of applicability are a useful filter for the selection of case studies. Looking at each area of applicability:

In support of performance standards: Simulation may be used to provide information that will satisfy approval bodies and others who may subsequently be called in to adjudicate should the performance of the design be less than required. In such work simulation needs to support standard performance assessments (e.g. summer overheating, condensation risk) and conform to standard quality assurance and reporting conventions.

In support of novel design: These designs propose the use of combinations of form, composition and control which are less well understood and which must be carefully assessed to ensure that the needs of the client are met. The design team may have an intuition as to how the design will perform or a suspicion that the design may be problematic. Because resources are less constrained and risk is more apparent, this class of problem has often been perceived as the domain of simulation.

In support of best practice: For some practitioners, best practice designs are a local vernacular—implying an intuition as to performance, established procedures, readily available information, etc. Such work requires care, attention, professional judgement and varying degrees of creativity, but it rarely ventures knowingly into untested waters.

Best practice work implies that the resources available for assessments are constrained, and often results in conservative approaches to design and engineering. If simulation is to be more widely used in this context, one would clearly wish to identify constraints in use. However, it may be that ease-of-use (which has been the focus of the work of many tool vendors), is actually less important than an understanding of how decision support tools can offer value for such pragmatic projects.

In support of research: Professional institutions and government bodies have made use of simulation as a research tool and to support the creation of simplified tools which are used to show conformance with prescriptive or performance-based standards. Here, simulation can be used in parametric mode to generate the underlying correlations or to define limits for the combinatorial possibilities allowed in a prescriptive standard. Simulation has also been used to test new constructions such as transparent insolation materials (TIM) or to identify the performance characteristics of emerging technology (e.g. facade mounted PV modules).

Over the course of this work the following project types have been used to observe, conjecture and test simulation:

- consulting projects
- European research programmes (including simulation based design studies, technology transfer initiatives and support services)
- simulation workshops as well as one of Europe's few simulation-based graduate and post-graduate programmes

Observations also include the work of others within the University of Strathclyde, and other research institutes and in professional practice. Five projects have been selected as case studies to explore:

- tool applicability within consulting practice, research and teaching;
- the pace and demands of the design process and the role of simulation within this process;
- issues within and outwith the tool which constrain design decision support;
- the benefits which might be gained from alternative approaches to the deployment of simulation and training of users.

The case studies often stretched the capabilities of both the tool and the practitioner; sometimes they required the evolution of new patterns of work, new assessment facilities or the extension of tool functionality.

The following is a summary of the case studies.

Detailed design study - Performance Assessment Service (PAS)

This was part of the ETSU Passive Solar Programme which explored the performance of passive solar designs via a series of case studies [Clarke 1989, BDP 1989, Hand 1991]. The author provided support to simulation teams both in terms of advice and in adapting the simulation tool to the needs of the programme. This work extended from February 1989 to December 1991. The Delta 100 office block, has been selected as being indicative of assessment demands and approaches taken.

Urban scale design study - Regensburg Solar Quarter

This was part of the EC's Solar House programme which explored energy supply and demand patterns in an urban regeneration project [Clarke et al. 1996, Fitzgerald and Lewis 1996] . Assessments included solar access studies and combined heat and power integration within housing blocks at several densities.

Consulting - Graham Hills Building Facade Study

Simulation is often called upon to assess existing building stock. This work was commissioned by the GA Group during 1992 to support a refurbishment study of a mid-sixties, 16,000 m^2 office block being converted to academic accommodation [GA Group 1993]. The brief required that simulation not constrain the pace of the project, and that staff of GA would be concurrently trained in the use of simulation. The case study thus looks at several issues: abstracting typical portions of the building, the rapid evolution of a simulation model, and the efficacy of using an active project as a basis for training.

Research into cooperative working

Traditional component-based approaches to design assessment are not well placed to deal with projects which rely on dynamic interactions between facets of a design. To assess how simulation might approach such a project, the author acted as a participant/ observer in a multiple-discipline design team attempting to optimise a design which incorporated both active and passive solar design elements [Hand et al. 1993]. Here simulation is used to optimise facets of the design and observe how a practitioner's understanding might be enhanced.

Training

This case study focuses on the precursor to simulation practice—the training of practitioners in the use of simulation as a virtual laboratory for exploring issues of building physics and systems design [Hand and Crawley 1997]. It is in such an environment that new generations of design professionals are equipped to use simulation and alternative approaches for training are tested.

The next five sections (3.1 to 3.5) review each of the case studies in terms of their objectives, approaches taken, models developed, principal findings and implications for the design of simulation tools.

3.1 Detailed Design Studies

Description and scope

As simulation has matured and has been seen to provide information which is both of value to the design professions and assists with high level goals (e.g. the reduction in CO_2 emissions), governmental bodies have become interested in initiatives to transfer advanced decision support technology into practice. For example, the Energy Technology Support Unit (ETSU) commissioned the Passive Solar Programme [Hand and Clarke 1992] to explore the performance of passive solar design strategies via a series of simulation supported design studies. The European Commission has supported a number of initiatives, e.g. the Solar House project [Fitzgerald and Lewis 1996], which focused on the integration of PV modules into buildings, Daylight Europe [Clarke et al. 1996] which focused on designs optimised for daylight utilisation, and the IMAGE project [Clarke et al. 1998] which was concerned with advanced glazing systems.

Such design studies are a potentially rich field for exploring the factors that hinder the use of simulation by the design professions. The Passive Solar Programme is used here to focus the discussion.

In 1988 ETSU initiated a programme of work in the field of passive solar buildings called the Passive Solar Programme (PSP) to "look at designs involving unheated, highly-glazed spaces such as atria and conservatories" [Hand and Clarke 1992]. As part of this programme, leading design firms undertook a series of simulation supported design studies in which existing office blocks and hotels were redesigned with passive solar features such as atria [BDP Energy and Environment 1989]. To support the design and simulation teams, a support service was established by ESRU and run by the author. Full details of the support service, its observations, findings and deliverables were made available in a final programme report [Hand and Clarke 1992]. In essence, the support service was charged to install simulation modelling within the participating design practices and to support its application thereafter. Several aspects of the PSP are relevant here.

- The PSP was overseen by a national research laboratory and managed by a lead contractor who subcontracted tasks to other geographically dispersed firms, consultants and research groups. Design goals were set by the lead contractor; the subcontractors deployed simulation towards these goals and generated the performance data which contributed to the final report. Such distributed decision making is not uncommon in design projects and appears to have a number of implications for design decision support.
- Each design study involved a comparison of the passive solar design with a conventional reference design. After initial findings the models evolved to address a new set of agreed design questions. At each stage, the state of the model and current performance predictions were abstracted for client meetings and interim reports.
- The programme allowed for coding interventions in response to coding and functional deficiencies.
- Staff with different backgrounds and interests needed to acquire skills in order to undertake their work. Some proceeded with little or no training, others learned during the course of their work and some were provided with formal training before they commenced their work.

One design study, the Delta 100 office block in the Delta Business Park, Swindon, Wiltshire, England (designed by The Oxford Architects Partnership, and completed in 1984) illustrates the level of complexity attempted within the PAS.

The office block, shown in Figure 3.1, comprises 4,800 m^2 of primarily open plan offices on three floors around a courtyard (14 m by 14 m) with the building rotated 45° from a cardinal orientation. Facades were floor to ceiling tinted glass with insulated spandrel panels. The building was air-conditioned with each floor separated into two zones in terms of environmental systems and controls. The PSP goal was to discover the relative performance of the building, as built, against reference designs which included an atrium, natural ventilation and critical placement of internal mass. Some of the design questions posed during the Delta 100 assessments were as follows:

- What is the magnitude of winter solar gain and how well can an atrium act as a buffer zone?

Figure 3.1 Collage of Delta 100 building.

- Can additional mass adjacent to the atrium store solar gains, and moderate tendencies to overheat?
- To what extent can an atrium enhance the natural ventilation and natural lighting potential of the building?
- How often and under what conditions will the atrium overheat?
- Can a redesign of the building enhance comfort without air-conditioning?

As the scale and complexity of the building was moderate, and both natural ventilation and the orientation of various sections of the building might influence comfort and environmental controls, it was decided to adopt a whole building approach as shown in Figure 3.2.

The results obtained from the study, as included in the lead contractor's report [Sluce et al. 1993], concluded that the redesign would provide a 44% reduction in energy demands without being unacceptable in terms of daylighting, overheating, space access or flexibility of use.

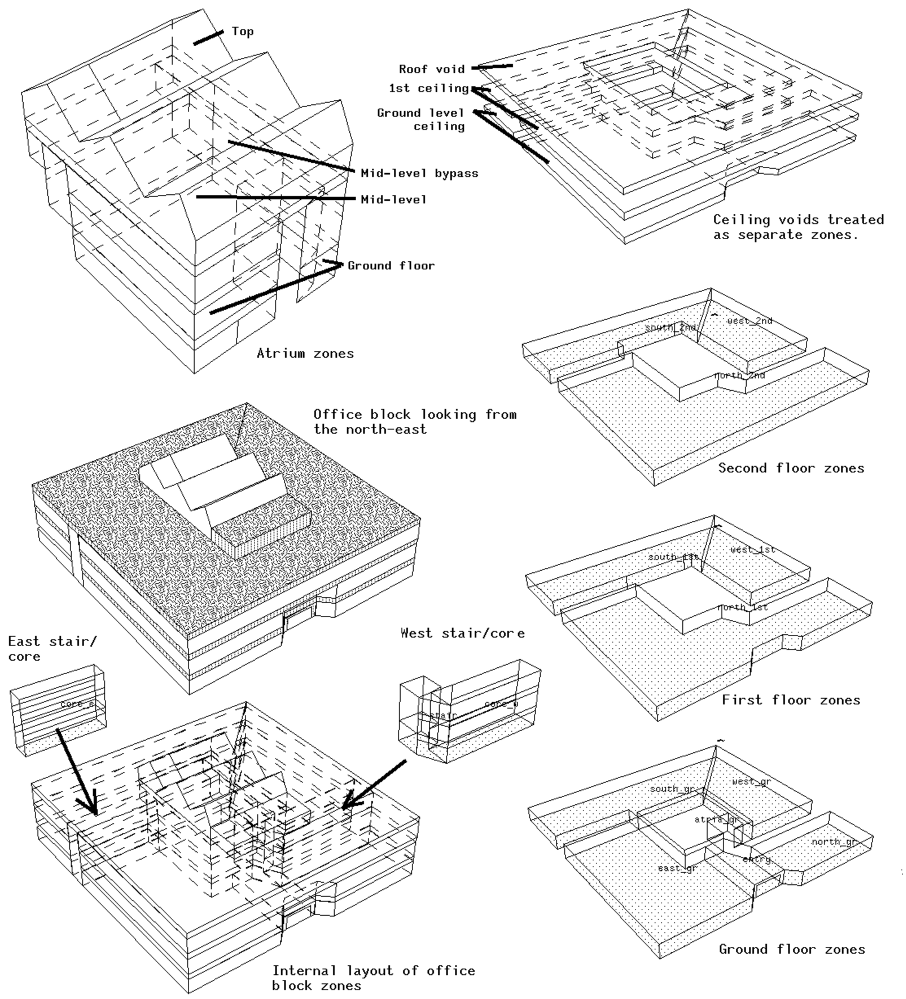


Figure 3.2 Collage of Delta 100 model.

Observations

Evidence from this and similar design studies showed the following:

- Detailed design studies invariably involve a number of participants, often disparate in terms of geography and expertise, who work cooperatively towards a stated goal.
- Project goals tend to be constrained by the competence of the contributing groups and their tools; design studies inevitably involve an evolution of skills and software.
- Each team contributes resources and expertise, but teams tend to work separately between formal meetings.
- Investigations, whether they progress in parallel or sequentially, are data-intensive and thus data-bound.
- Initial planning often underestimated the resources required to train staff, to manage the volume of information contained in models and performance predictions, to support model exchange and

dissemination of findings and, critically, to support model evolution.

- Beyond a certain level of complexity, the attention needed to maintain notes and ensure internal consistency and inter-relationships increased significantly.
- Only those who were experienced in the tool were well placed to understand the composition of models or to access information about them.
- Considerable mental agility was required to shift attention between models and then make incremental refinements.

Thus, the attainment of project goals, and the use of simulation to support such goals, depends on the co-ordination of tasks, intra-team communication and data exchange, as much as the theoretical robustness of the tool.

Issues arising

Initial planning in the PAS design study allocated 28 person-days to assess a conventional design and 34 person-days for the novel design variants—a considerable resource in either a research or consulting context. In the event, only four of the intended twelve designs were completed. A number of the causes of this shortfall were related to the implementation and application of simulation [Hand and Clarke 1992]:

- Simulation presumed the user's full attention, especially as model complexity increased.
- Descriptive tasks tended to be optimised for the needs of the computational engine rather than to minimise ambiguity for the user or to support the rapid and transparent exchange of models between participants.
- Facilities were geared for the creation rather than the evolution of models.
- The simulation data model held a subset of project information and it was assumed that the balance (e.g. related databases, assumptions, simulation parameters) was managed by the user.

Altogether, the environment was unsympathetic to shortcomings in the user's attention, to uncertainties in (or absence of) supporting data, or to the pace and evolving nature of assessment tasks. Prior experience with DEROB and BLAST at the National Building Research Institute, Pretoria, as well as discussions with those involved with DOE-2 and Power-DOE [Crawley 1998, Crawley and Lawrie 1998] indicates that such observations are true for other tools of this class.

Design studies also raise issues as to how simulation is used (whatever its design and facilities).

- Deliverables suffered when a whole building approach was applied where a focused study would have sufficed. For example, hotels are characterised by small rooms and a high ratio of internal partitions to external facade, but a whole building approach required the omission of most internal partitions.

- For operational reasons, some models were created prior to the identification of design issues. As new information became available, it was difficult to adapt the model. Knowledge is required on when to delay model creation. Tools need to give better support for incremental approaches to model creation.

Design studies were also constrained by the volume of information to be maintained and the need for quality assurance—particularly where several projects were active, but at different stages of completion. Under such circumstances, weaknesses in quality assurance procedures, lack of clarity in tool interactions and lucani in the data model have all been seen to compromise project deliverables and to increase the resources required to carry out assessments. Even where investigations do not overlap, such issues reduce the efficacy of simulation.

Where design studies involve a physical separation of those who plan/ manage a project and those who carry out the assessments, a number of issues arise:

- Even when project goals are explicit, assessment teams may evolve inappropriate work practices or simulation models when they work in isolation. Opacity of models and assumptions delays the identification of problems just as much as it obscures potentially useful performance patterns.
- Where performance predictions need to be shared between participants, understanding depends on the ease and transparency with which models and details can be shared between participants. The case studies indicate that simulation makes it difficult for those who are not *au fait* with a model to recover relevant details.
- Where an initial approach proves inappropriate or indicates a change in focus or level of detail, the design studies indicate that project deliverables can be compromised by barriers to the implementation of alternative approaches.

Implications for simulation

- There is a need for simulation to be open to casual (i.e. intermittent) use, without the need for prior knowledge of simulation models, and to present its information concisely, unambiguously and in terms which are familiar to the user.
- There is a need to accommodate the considerable compositional and operational complexity observed in the built environment.
- Simulation must allow the focus of assessments to evolve over time—e.g. from system capacity assessment, to glare discomfort evaluation.
- Descriptive tasks need to be better managed. For example, assumptions and component details which may be critical to the understanding of the model should become a part of the data model.

3.2 Urban scale design studies

Description and scope

As well as supporting design decision for single buildings, simulation can also support urban policy initiatives which are often concerned with a mix of detail and macro-scale questions. In the early 1990's utilities explored how they might become "energy service suppliers" and often found demand-side management more cost effective then extending capacity [Selkowitz et al. 1992]. Currently, electricity power utilities in the UK are concerned with how distributed small-scale and renewable energy sources can be integrated into the electrical grid. This has implications on both demand-and supply-side management. The information required for such studies can be expensive to obtain via large scale monitoring or is not available at a suitable frequency: simulation can be used as a *virtual metering* system.

The possibility of managing delivered energy at the urban level requires, among other things, that the patterns of demand be known for the anticipated mix of buildings and population demographics. To explore what might be required from simulation to address macro-scale issues, a project involving simulation support for the design of a "Solar Quarter" in Regensburg, Germany (a project by Sir Norman Foster and Partners with Professor Thomas Herzog, Renzo Piano, Sir Richard Rogers and Norbert Kaiser) has been selected [Clarke et al. 1996]. This project, undertaken as part of the EC Solar House Programme, was concerned with energy autonomy and solar access on Unterer Wohrd island (see Figure 3.3) in Regensburg.

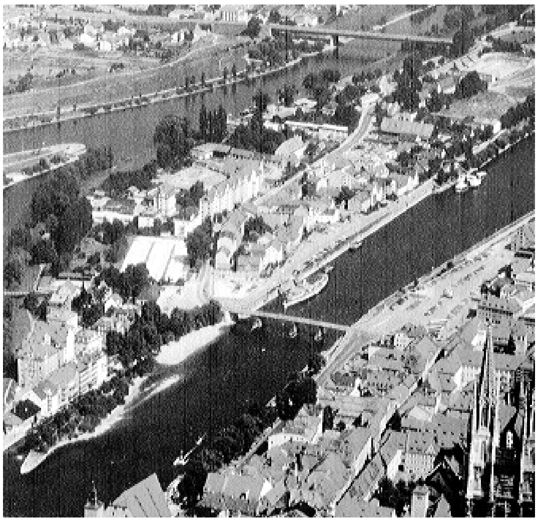


Figure 3.3 Unterer Wohrd Island, Regensburg.

Several design issues were considered. One was the density and placement of housing for the optimisation of solar access to enhance natural light, maximise winter solar gain and minimise lighting/ heating energy demand. Another issue was how to make use of information technology to match supply with demand across the different energy use sectors and seasons.

The approach taken was to employ a directed search rather than an extensive (speculative) parametric traverse. In the Regensburg study the initial questions related to general building massing and solar access. This was treated as a visual assessment of rough models with different dimensional characteristics, two of which are shown in the upper part of Figure 3.4. The metric was shadow patterns at representative times of the day for representative seasons (shown in the lower portion of Figure 3.4). From this the design team arrived at rules for the aspect ratios of buildings. A similar process was followed to arrive at rules for the design of courtyards and to identify locations with constrained solar access. In a parallel study, typical housing unit plans were reviewed and possible design variants discussed. Several of these were abstracted into models and short period simulations were run to support design decisions on fenestration types, orientation, and construction. One such housing model is shown on the left of Figure 3.5.

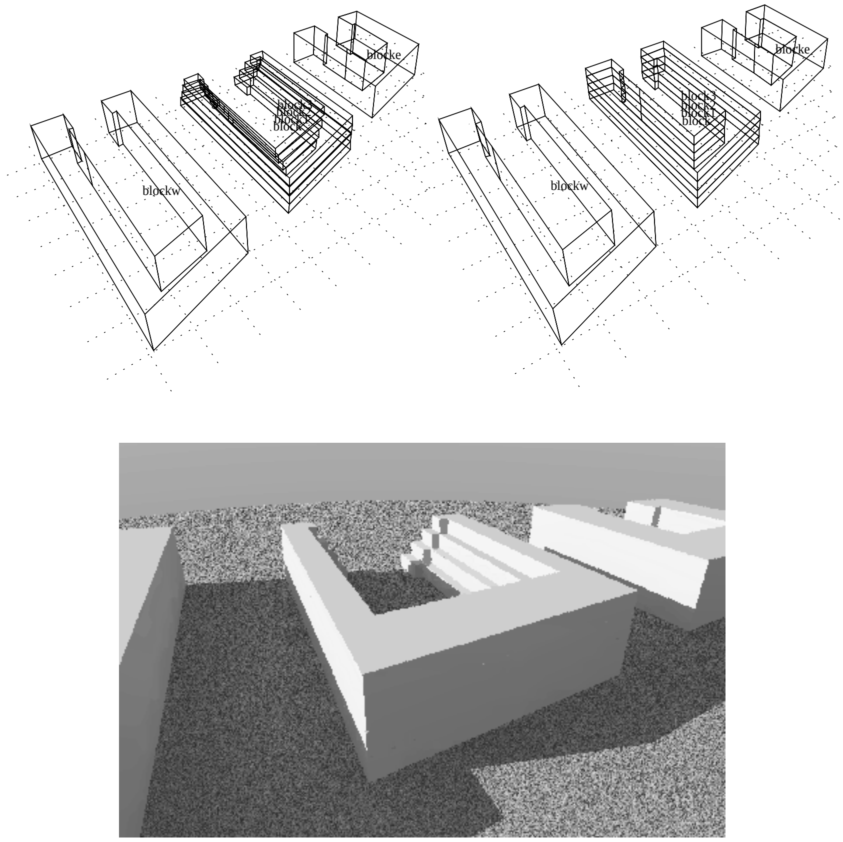


Figure 3.4 Solar access models and shading plots.

Information from these initial simulations was then drawn together to support a second study focused on building massings and operational variants. One such model is shown on the right of Figure 3.5. The pattern was to add geometric detail or an operational variant to the set of models only after these had been tested in a focused study.

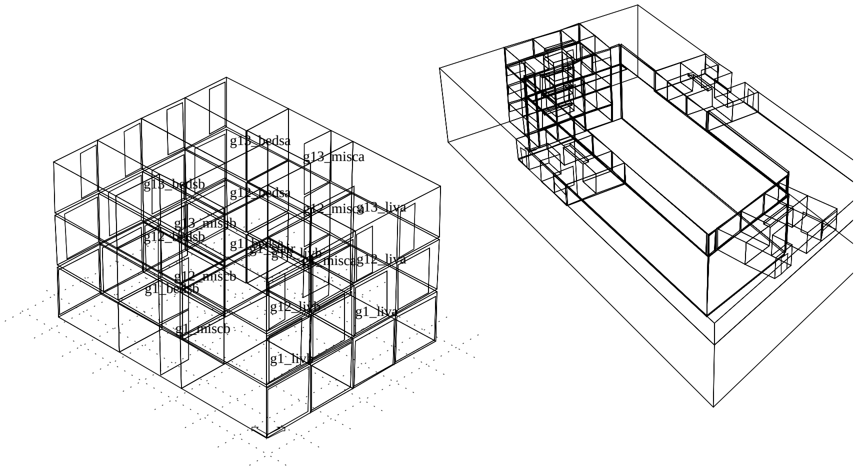


Figure 3.5 Housing models.

The results of the study were then aggregated into graphs showing the heat to power ratio for 36 design variants as shown in Figure 3.6.

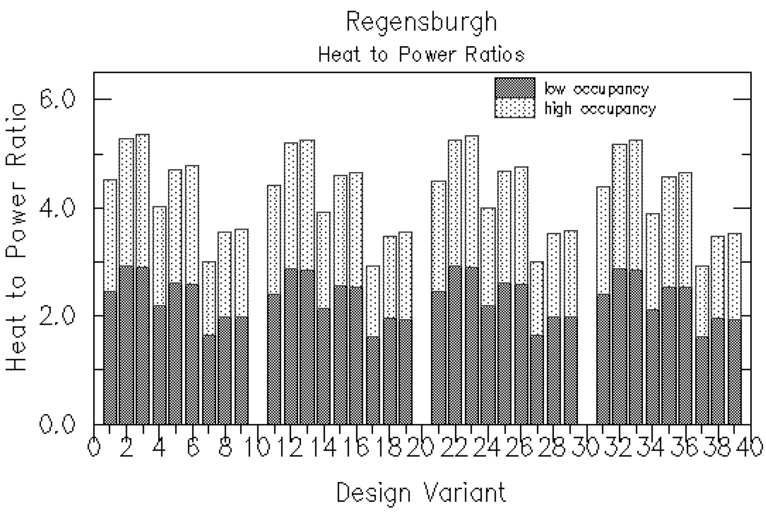


Figure 3.6 Performance of design variants.

The impact of each design variant on the hourly demand pattern was also determined. The results, such as those shown in Figure 3.7, were used to answer design questions such as:

- Will high quality windows reduce the peak load and energy demand?
- Will any of the proposed mixes of light- and heavy-weight construction types smooth the demand for energy over the day and so shift demand to reduce peak requirements?

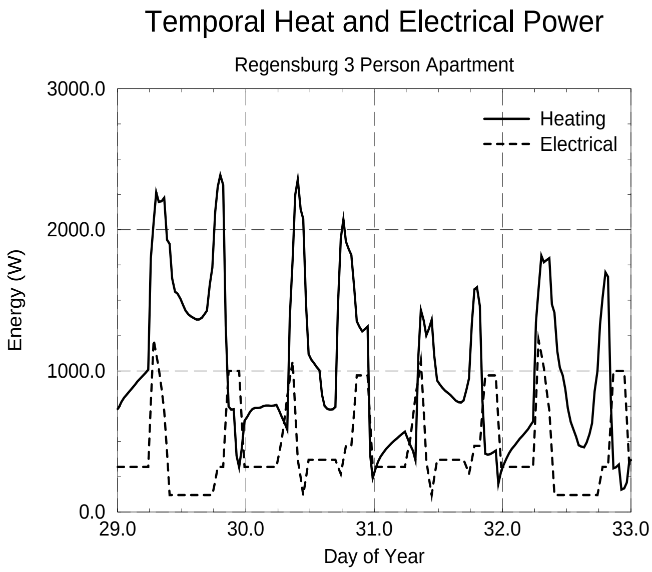


Figure 3.7 Demand profile for one housing type.

Observations

Projects of this type require information at a number of levels of detail and with sufficient design variants to be representative of urban scale demand profiles. There may be several metrics and design questions to be addressed. The practitioner must balance initial investigations which seek to identify overall patterns, studies checking sensitivity to specific design variants and setting up and running a set of models whose performance can then be conflated and scaled. Rushed initial studies can easily lead to inappropriate detailed studies and muddled results.

Where a project requires a number of model variants, care in initial planning and the tactical use of simulation facilities can identify useful and not-so-useful approaches, limit needless redundancy and reduce the resource required for error checking. The latter can be especially time consuming (e.g. confirming parameters, ensuring contiguity, ensuring that each run can be replicated if necessary). Kaplan et al. [1992] found a significant source of discrepancies in savings estimates was from insufficient consideration of baseline models (without energy conservation measures) in comparison to the attention given to as-designed models. Such tendencies could compromise an urban scale study.

Issues arising

The discovery of performance patterns within a group of models is complicated by a number of factors. Firstly, there is likely to be a wealth of data, and without clear opinions as to the metrics of the project critical indicators can be lost. Second, causal links are difficult to maintain in parametric studies—the change in performance may be subtle and some time may have elapsed between the setting up of the design variant and the viewing of results. Third, some practitioners persist in reviewing only high level performance indicators when critical patterns may only be understandable at greater detail. Lastly, models which lack clarity (or incomplete documentation) can lead to confusion as to the specific combination of design variants.

Implications for simulation

Support for urban scale design would be enhanced if practitioners could:

- incrementally add and remove geometric detail;
- rapidly (and robustly) change model composition (e.g. change all skylights to clear double glazed units);
- rapidly (and robustly) replicate and relocate a base case (housing) unit to form sets of units (each supporting a specific design parameter);
- have guidance on successful tactics for tool use and naming schemes which limit error propagation;
- access model reports which clearly identify the differences between models.

3.3 Consulting

Description and scope

Both practitioners and developers recognise that consulting is constrained in terms of time and resources while being prone to rapid changes in focus. Consulting projects (some completed in days, others requiring months) have been undertaken and used to test whether it is possible to deliver simulation within the real time, cost-constrained context of consulting. It was possible to observe sequences of information exchanges within design teams and to alter computational facilities, forms of interaction or the underlying data model in response to project demands and observe the impact.

Appendix C includes details of several consulting projects and one of these, undertaken under the auspices of the Energy Design Advice Scheme (EDAS) [GA Group 1993], is described here to typify the process.

In 1991 the GA Group of Glasgow undertook a refurbishment feasibility study on the 16,000m^2 Graham Hills Building as shown in Figure 3.8. The refurbishment of the 25 year old Glasgow building was driven by the need to update the building facade, adapt its internal space planning, and address a history of poor thermal, visual and acoustic comfort. The design team, which included ESRU, set out to study alternative facades, internal layouts and services designs to support office accommodation in three cost ranges. Each design variant was to be compared with the performance of the existing building.

Several aspects of this project commend it as a case study:

- Simulation was included from the outset of the project and therefore it was possible to influence the design team as the project evolved.

Figure 3.8 Graham Hills building (viewed from south-east).

- No concession was made to the time frame of the study to accommodate simulation tasks.
- Design questions involved the overall form of the building as well as the implications of facade details and space planning on comfort, natural ventilation and running costs.
- The design team was interested in acquiring simulation skills and seconded staff to be trained within the project. This allowed a close observation of how trainees interacted with the simulation tool and how the simulation tool assisted or constrained their work.

The initial task of the design team was to identify areas of the building requiring attention and then use a series of performance studies to guide the specification of subsequent design details. Possible outcomes were that facade details might need to change depending on orientation and exposure and mechanical ventilation might be required in some rooms.

The challenge was to represent the essential characteristics of the facade and internal layout to a level of detail which would allow the impact of design variants to be assessed. Before proceeding with modelling tasks, it was necessary to consider the nature of design questions, the metrics used to judge performance, and the pace of the project.

The need to rapidly address several detailed (but as yet unspecified) facade treatments and room layouts made a concise approach essential. The complexity of the plan form (e.g. several courtyards, a sloping site, different solar and wind exposures) argued for a whole building approach. However, a whole building approach implied a degree of abstraction which was inappropriate for facade design studies. At the other extreme, concentrating on one or two offices risked missing or misjudging performance metrics because the relationships within the mix of cellular and open plan offices and facade exposures only became evident at the scale of a wing of the building.

The approach taken was to compose an initial solar model to explore patterns of solar access and summer overheating potential. A series of composite models, as shown in Figure 3.9, were then developed which captured the principal accommodation types, adjacencies and facade orientation/ exposure.

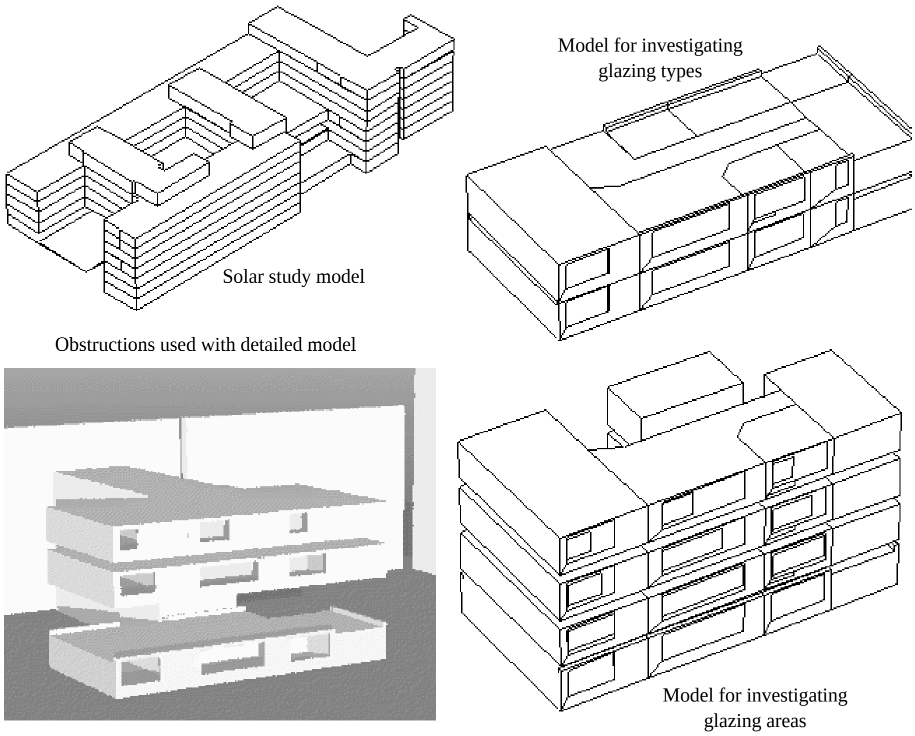


Figure 3.9 Solar exposure and detailed models.

In order to deal efficiently with design variants, this arrangement was replicated on additional levels. For example, the model in the lower right of Figure 3.9 applies a different area of glazing to each level, while the model in the upper right applies a different type of glazing. Other models were adapted from this initial model to deal with levels of insulation and the like.

Because the basic model was composed to allow a rapid implementation of such changes, responses to "what if" questions were often delivered on an overnight basis even though such a quick response was unusual for a project of this complexity.

Observations

More often than not, the delivery of useful information to the design process was seen to be related to efficacy of approach rather than to computational speed. Attention to the nature of the design questions being posed and clarity in the metrics selected helped to avoid needless complexity and guide the creation of models. Other observations are listed below.

- Practitioners tend to search for performance patterns which confirm an initial hypothesis before proceeding with detailed investigations.

- The focus of attention and the level of detail often shifts abruptly.
- In consulting, models evolve, and anticipation of change is critical to the planning and execution of simulation tasks. Such changes need not require additional resources if the underlying method of the study is sound.
- Models tend to evolve from compositions characterised by place-holders (defaults) to models substantially specified.
- Information related to models and projects held outwith the simulation environment is problematic in terms of quality assurance.
- Models which are difficult to understand (i.e. convoluted, tersely named, sparsely documented) are inappropriate in a consulting context. It is a false economy to rush the descriptive process.
- Even with a rich set of reporting functions, practitioners are often ill-equipped to use native reporting functions to explore and understand performance issues.

These tool deficiencies had as much to do with issues of clarity, consistency, logical naming schemes, availability of information and persistence of information as with productivity aides, automation and the form of graphical user interface.

Issues arising

While most buildings are "created as one-of-a-kind designs, where reinvention of the solution is the norm" [Selkowitz et al. 1992], there is still uncertainty associated with how such a collection of components and systems will perform. Thus the consulting process is characterised by the practitioner’s search for performance patterns which confirm an initial hypothesis before proceeding with detailed investigations. To use a bird-watchers terminology, consulting work has a different "jiz" (the recognition of an entity by a subset of indicators such as form, behaviour and context). Thus simulation must help the practitioner to understand the nature of the design in terms of early indicators as well as in its support for detailed investigations.

The need to support understanding (and perhaps impress a client), influences the nature of models. If this results in clear and concise representations of the design, then the influence is a positive one. However, if it encourages the creation of overly elaborate models (i.e. to mimic CAD) then this inevitably detracts from decision support.

Consulting is also typified by the demands to explore a range of design options. This becomes problematic when expressed in terms of the annual energy implications of four glazing types, three occupancy regimes and three temperature setpoints. Such combinatorial explosions have quality assurance and reporting implications which the current work addresses. However the true problem with such approaches is that they are not directed searches and critical performance indicators can be lost in the volume of reporting. The demand for a full traverse of all variants also precludes potentially useful exploratory studies and delays feedback to the design team. Guidance on the selection of design options is a necessary adjunct to coding modifications which clarify and regularise parametric tasks.

Two reporting issues emerged. Firstly, the graphs and tables traditionally presented in formal reports had to be obtained by exporting information from the interactive environment, increasing the resolution and adding annotations. Where this involved extracting sequences of data to be held in separate files, the time and quality assurance resources increased considerably. Secondly, the strictures of formal reporting do much to limit the design decision support potential of simulation when compared with the depth and breadth of interactive views of performance data.

Implications for simulation

The consulting case studies provided several pointers for the evolution of simulation:

- simulation must evolve to better tell *the story* of the design to the simulationists and design team;
- audit trails (e.g. decision points, assumptions and problem variants) warrant consideration as part of the data model;
- work practices and tool facilities need to support the rapid evolution of models;
- guidance is needed on alternative methods which include exploratory studies and directed approaches to parametric studies;
- quality assurance is central to consulting, but the resource needed to undertake this task is almost always underestimated;
- calculation processes should not be silent—an evolving display of selected performance metrics can provide valuable early indicators;
- reports and graphs which are designed for interactive decision-making tend not to be optimal for formal reporting because of their lack of resolution and terse annotation.

In response to the above, it was conjectured that it was necessary to enhance the documentation associated with simulation models, in particular the logical naming of descriptive entities, the recording of the intent of the exercise and the documentation of assumptions. It was also necessary to investigate how the interactive display of performance data can better meet the needs of design teams and how simulation can be used as an on-line tool rather than a back-room activity.

3.4 Cooperative working

Description and scope

Rittelmann [1995] observes that engineers (as specialists who often use systematic and linear processes) and architects (generalists who often use lateral and holistic approaches to problem solving) were more likely to work cooperatively when they shared a common CAD tool. Rittelmann also noted the increasing relegation of the engineer to "that of a technician" in projects where mechanical solutions are prescribed (e.g. use of packaged air handling equipment). Observations from a recent survey [Donn 1997] indicate that such "technician" roles are all too common and has implications for design teams understanding the performance implications of design decisions.

Simulation offers the possibility of studying interactions between facets of a design which traditional (i.e. component-based) engineering approaches do not allow. For domain experts to work cooperatively to define, test, evolve and learn from simulation models, a number of issues must be addressed. It is necessary to identify methodical approaches to cooperative tasks, the skills and resources required, the nature of the constraints to cooperative working and the implications this has for the design of simulation tools.

In order to explore these issues, the author supported a project with a design team comprising domain experts for passive solar design, mechanical engineering and plant systems modelling [Hand et al. 1993]. The goals of the case study were to observe the interactions between the participants, the nature of the information exchanged, the process of arriving at a consensus on how to approach the project and what was required to understand the resulting performance predictions. The author's role as participant/ observer allowed the close observation of both the design process and the use of simulation.

The design project, shown in Figure 3.10, included a mix of passive and active solar design elements and employed a different heat delivery mechanism in each room (e.g. direct solar gain with an oil filled radiator backup, a solar sourced wet central heating system with a radiator and a fan coil unit, and an air-based solar collector). The brief was to optimise the whole of the design and where possible, to do this via integrated assessments.

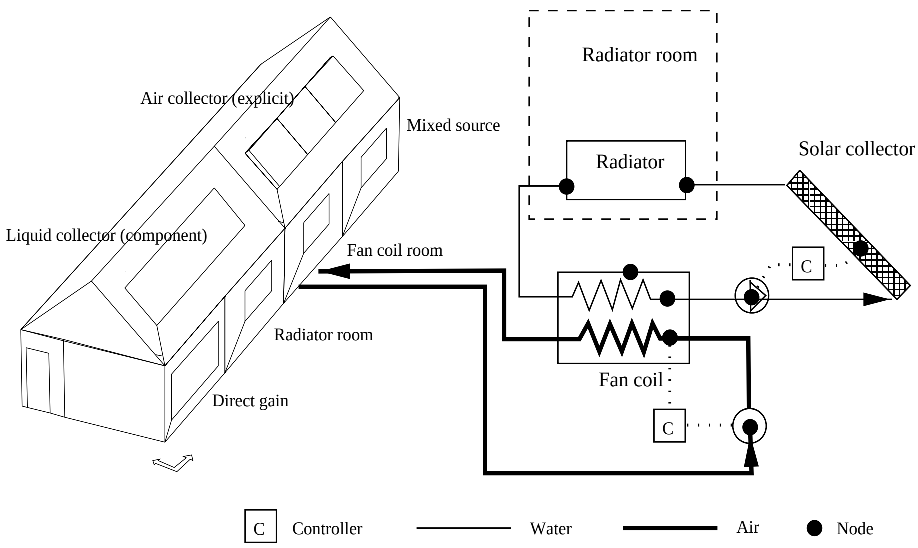


Figure 3.10 Building, flow and plant details.

The approach taken by the design team was to come to an initial agreement on performance metrics (e.g. specific temperatures and flow rates), rules for abstracting the design (e.g. the need for additional geometric detail in the direct gain room and the importance of the temporal response of the oil-filled radiator), the underlying methods to be used (e.g. whether air movement required a network approach and whether the solar collectors would be treated as plant components) as well as the nature of the assessments to be undertaken (e.g. analysis time frequency and period). The building, flow networks and plant components were then composed and checked separately. The team then met to link the model elements and proceed through several cycles of simulation, analysis and fine-tuning as the performance of each domain, and then the whole design, emerged.

Observations

The descriptive process proved straightforward for the building and air network, but raised particular questions as to the representation of the heating system components. For example, an ideal control would be easy to set up, but would not account for the response time of an oil-filled radiator and a plant representation of an air-based solar collector was ill-suited to integration within the roof.

During this case study:

- experts were exposed to aspects of the data model with which they were unfamiliar (e.g. the plant expert was not used to the patterns of building/ system feedback and the passive solar designer was unaware that the systems side required timesteps of a minute or less);
- thermophysical relationships were found to be more explicit than would have been the case with a component-based assessment;
- the process of describing interactions between domains was a source of confusion;
- maintaining inter-domain relationships required a greater resource than the design team had anticipated.

Issues arising

Although it proved unnecessary for the experts to grasp the nuances of other domains, an initial joint understanding of how the design might work and the interactions between domains was required. This was achieved verbally and on paper, but some issues did not become apparent until model creation and others did not arise until the testing phase. Some iteration appears to be required in model development.

The time taken to develop confidence in the model and in the resulting performance predictions was greater than anticipated. The design team initially attempted to prove each facet of the model separately and quickly switched to a team approach because the tight coupling of thermophysical processes produced patterns which could not be understood in isolation.

In a cooperative project, diverse information sources and links, ordinarily maintained by an individual simulationist, needed to be managed explicitly by the group. This was a source of frustration and considerable misunderstanding. Questions such as "where are we with this project" were not unusual.

Implications for simulation

The implications for the design of simulation tools are:

- unambiguous feedback is needed as to the form, fabric, networks and components which constitute the model;
- facilities are needed to support understanding of detailed energy flux and fluid flow patterns;
- simulation does not explicitly recognise team working or enable the manager of a cooperative project to monitor progress or manage tasks;
- new views of performance are required to cope with tightly coupled problem types.

One of the least appreciated aspects of the information demands of thermal simulation is that after such a multi-domain model has been composed, it encapsulates much of the descriptive information used by other tools in the design process. Clearly, no single tool is capable of supporting the design process. The possibility exists for simulation to become a common focus point of discourse for the design team and a repository for project information.

3.5 Training case studies

In each of the case studies discussed thus far there were instances where a lack of skills or a critical misunderstanding of some aspect of simulation constrained the use of simulation. Thus, understanding how novices become simulationists, how professionals can evolve methodical approaches to complex assessments, and how simulation can better support the acquisition of such skills deserves careful consideration. The case studies which follow look at skills acquisition from the perspective of formal training (e.g. courses, workshops), informal training (e.g. visits to development sites and within the context of research projects) and project based training with a mentor.

3.5.1 Formal and informal training

Description and scope

The University of Strathclyde has proved a fertile ground for exploring simulation as an instructional medium. Each semester a mix of students studying Architecture, Building Design Engineering, Environmental Engineering and Mechanical Engineering at the undergraduate and postgraduate level use simulation within their course work. It has been particularly instructive to observe the issues which arise when simulation is introduced at progressively earlier points in a degree structure. Over time, such observations have provided information on alternative routes to the acquisition of skills.

By the book

Access to a workstation, manual and sufficient time to sort things out by trial and error are perhaps the default mode for those remote from the tool developer. Historically this form of skills acquisition has been associated with frustration on the part of the user as well as inefficient deployment of tools: this has been confirmed by observation.

Informal tuition

Those wishing to work on joint research programmes or to explore complex issues sometimes arrange intermittent access to a tool developer or expert to enhance a self-instruction regime. The University of Strathclyde has hosted a number of visiting professionals and researchers and such visits have provided ample opportunity for observation, conjecture and testing.

Formal tuition

Lectures and workstation sessions, supported by course notes, exercises and tutorials, are a traditional mode of skills acquisition in universities, and a number of these have been mounted at the University of Strathclyde. Table 3.1 is a lecture schedule for a typical introductory course on ESP-r with three hours of workstation sessions per week interspersed with two hours of lectures, discussions and demonstrations.

Table 3.1 Course lecture schedule.

Task	**Description**
First week	Overview of course contents and assignments.
Second and third week	Energy/ environmental monitoring, targeting and prediction.
Fourth week	Introduction to simulation-based modelling.
Fifth week	Capabilities of computer modeling. Demonstrations of thermal, lighting and acoustic programs.
Sixth week	Overview of ESP-r: strategies, approaches, and technical issues. Initial workstation session.
Seventh week	Theoretical basis of ESP-r: building side.
Eighth week	Theoretical basis of ESP-r: HVAC side.
Ninth week	Exemplar applications of ESP-r.
Tenth week	Future developments.

Note that workstation sessions/ laboratories do not begin until the 6th week of the course. Typically, one lecturer and one graduate assistant attend each laboratory of 20 - 25 students to check progress and answer questions. Students submit progress reports and assignments (including completed models) by electronic mail and, in the latter stages of course development, used hypertext-based tutorials and reference materials. Observations have focused on the nature of the skills acquired, the resources needed for keyboard skills and fundamental issues, and the format of exercises and notes appropriate for participants of different backgrounds.

Workshops

Workshops are a concentrated form of formal tuition often used for initial, advanced and project focused training. Workshops have been held for groups of mixed background and interest, groups who held common goals and for graduate and undergraduate students. Usually these were intense introductory courses for professionals and researchers, although advanced workshops have also been given. Table 3.2 is a schedule for a typical three day introductory workshop on ESP-r, with workstation sessions interspersed with presentations (marked with a P), discussions and demonstrations. Workshop observations focused on the balance between instructor interaction and support within the tool (e.g. context-sensitive help), as well as the pace and ordering of topics.

Table 3.2 Workshop schedule.

Task	Description
First day	
9:00 welcome and overview	Participants and course staff introductions. Defines the goals of the course and provides an overview (via demonstrations) of the essential characteristics of the tool. Later, this section was extended to discuss the "ethos" of the tool, a synopsis of core product model entities and what constituted simulation models.
9:30 exercises + assignments	Topics related to workstations, essential operating system commands, use of the window manager and essential applications such as email (to report progress).
10:45 exercises + assignments	Topics related to invoking modules, browsing existing models and commissioning simple assessments.
13:30 (P) background and overview	Having sampled the simulation environment, participants are given further background into the tool and an overview of its applicability.
14:00 exercises + assignments	Participants explore existing models of increasing complexity and how models can be extended to include shading, air flow and control of environment systems.
16:00 exercises + assignments	Participants prepare to compose a simple model, some continue work in evening.
Second Day	
9:00 (P) modelling approaches	Defining simulation tasks and model detail needed to support design questions.
9:30 exercises + assignments	Participants proceed with the composition of a simple model, optionally using CAD.
10:45 exercises + assignments	Topics relate to environment systems control.
13:30 (P) verification and validation	Participants' attention shifted to non-tool specific issues.
14:00 exercises + assignments	Participants add network air flow to their model and explore how this alters performance.
15:45 exercises + assignments	Participants add additional zones to their model and explore intra-zone heat transfer issues.

Task	Description
Third Day	
9:00 (P) current developments	Participants’ attention drawn to a range of topics which they might wish to explore later in the day.
9:30 demonstration: current developments	Instructors give concurrent presentations of advanced facilities.
11:15 exercises + assignments	Participants work on topics of their own choosing.
13:30 (P) simulation in practice	Mix of demonstrations, presentations and discussions.
14:00 exercises + assignments	Participants work on topics of their own choosing.
16:45 course review	Especially useful for feedback from participants.

Observations

The principal observations of skills acquisition modes follow.

By the book:

Self-learning only occasionally works. One reason may be that the most common source of information (the manual) is often written:

- as if there were no gaps between the developer’s view of what is "perfectly clear" and the users’ impression of chaos and visual clutter;
- to impress the prospective user, often at the expense of guidance on the management of simulation teams and simulation projects and the limits and caveats which are associated with the tool;
- including either overly simplistic or complex examples without mention of their rationale and assumptions (i.e. the very things needed for others to attempt similar work);
- in a manner which is more adept at presenting the "what" but awkward when it comes to the "why".

Informal tuition

The addition of intermittent access to an expert to either self-instruction or formal training regimes has been observed to be a viable technique for those who wish to add new facilities or explore a particular facet of simulation. Informal tuition demands self-motivation on the part of the visitor and the ability to adapt to the host’s environment.

The case studies identified a number of limitations. Those attempting code extensions are constrained by the documentation and structure of the code, timely access to experts and the need for incremental revision and testing. Those exploring a facet of simulation often find that example problems or documentation are unclear and that facilities are immature. This has resulted in a number of refinements to coding style, documentation and testing regimes which is discussed in Appendix A.

Formal tuition and workshops

Three and four day workshops have been successfully used to explore intermediate topics as well as to provide intensive introductions for professionals. Observations show a considerable variation in time taken to complete exercises. For example, one exercise was to take a standard wall construction and alter its thickness in a database and then associate this new component with a half dozen surfaces in two zones. This required accessing, editing and updating a database, changing focus to each zone in turn, selecting relevant surfaces and associating the name of the revised component with them. A skilled user would complete this in a matter of minutes. Those with previous use of simulation claimed the task took twenty minutes while students required up to five hours. To explore the variation in progress, students were asked to keep a log of the time spent on various topics and exercises as well as notes on sources of information. This is discussed further in Chapter 8.

Allowing participants to progress at their own pace through a standard set of tutorials and exercises makes considerable demands on course staff, but allows those with prior simulation experience to progress quickly and to spend more time working on advanced topics. Usually one or two participants progress rapidly. A similar number who inevitably arrived with few, if any, workstation skills tend to struggle.

The case studies show that the rate of progress in both workshops and formal tuition is sensitive to the order and pace of topics. It is a false economy to approach workstations before participants are given an overview of simulation, especially if they are novices who are not in a position to recognise what is being presented on screen or to undertake basic workstation tasks (e.g. moving between folders, copying files). Participants usually do not have sufficient attention to cope with learning both basic operating system commands and simulation within an introductory course and so they should acquire basic skills prior to undertaking the course.

Issues arising

For those with simulation experience, the task of gaining skills in a second tool is complicated by a number of factors.

- The meanings associated with words and entities within the simulation environments differ—sometimes in subtle ways.
- Underlying methods differ, changing the emphasis or even the ability to assess particular performance metrics.
- The underlying data model differs, perhaps requiring the use of a weighting factor for the representation of thermal capacity in one case and an explicit representation in another.
- The assessment domains differ, perhaps including network air flow in one case and scheduled air flow in another.

All of the modes of skills acquisition discussed in this Section included some form of tutorial which used structured exercises to codify tasks and mark progress. Initially, these were composed for participants with any background (e.g. mechanical engineering undergraduates, architects and researchers) and covered a range of progressively more difficult simulation topics to ensure that both novices and more experienced practitioners had the opportunity to explore related facets of the tool.

Some ESP-r exercises take the form of "take the low energy office model and change all instances of construction X to construction Y and report how this alters the demand for heating energy". The progressive nature of the exercises ensures that a prior exercise covered how to acquire a copy of the model as well as the concept of geometry attribution. It would also be assumed that the invocation of a simulation would be routine by this point but that a reminder was needed to check both the heating capacity and demand profiles.

Donn [1995] describes a similar sequence of "Hot Box" exercises for the simulation program SUNCODE. Over a two week period, students progressed from a model of an unoccupied room and incrementally enhanced the resolution of the model and tested alternative components and building use. The students were asked to discuss and contrast the performance predictions at each stage and the influence of occupancy/ usage patterns. Donn found that students had a greater appreciation of the complexity of thermophysical processes after the progressive exercises.

Although progressive exercises support many aspects of skills acquisition, the context of their use in workshops and formal tuition and the issues of model syntax and tool navigation that they tend to focus on does not deliver the expertise needed to undertake design decision support. Exercises are often built around designs such as that shown in Figure 3.11. Many novices learn basic interaction skills by re-creating such models from plans and specifications. Those who are used to other tools use such models to establish a mental mapping between tool facilities and data models. Such models provide a convenient reference for the syntax of the tool and for entity relationships. They are not reasonable templates for use in professional practice because their composition was intended primarily to demonstrate syntax rather than having evolved from the demands of design decision support.

To go beyond issues of syntax and tool navigation the novice requires access to a range of models which address design issues and are of realistic complexity. If such models could be browsed and basic assessments carried out without prior knowledge of their contents, they would provide an early and powerful influence for those who were willing to explore their composition.

Implications for simulation

There is a historic tendency in workshops to get participants "on to the keyboards" as quickly as possible. This forces participants to learn rote patterns or adopt syntactic keys (unthinking translations of visual clues into a sequence of actions) which are difficult to acquire and remember. When novices are asked to compose models, they inevitably rush to the keyboard, and undertake impromptu compositions of needless complexity. Rather, learners should be required to undertake a planning exercise on paper to establish the essential characteristics of the design, the nature of its composition and appropriate level of detail.

A balance must be maintained between allowing novices to experience initial successes and the creativity enhancement which comes from an understanding of the underlying data model. This is

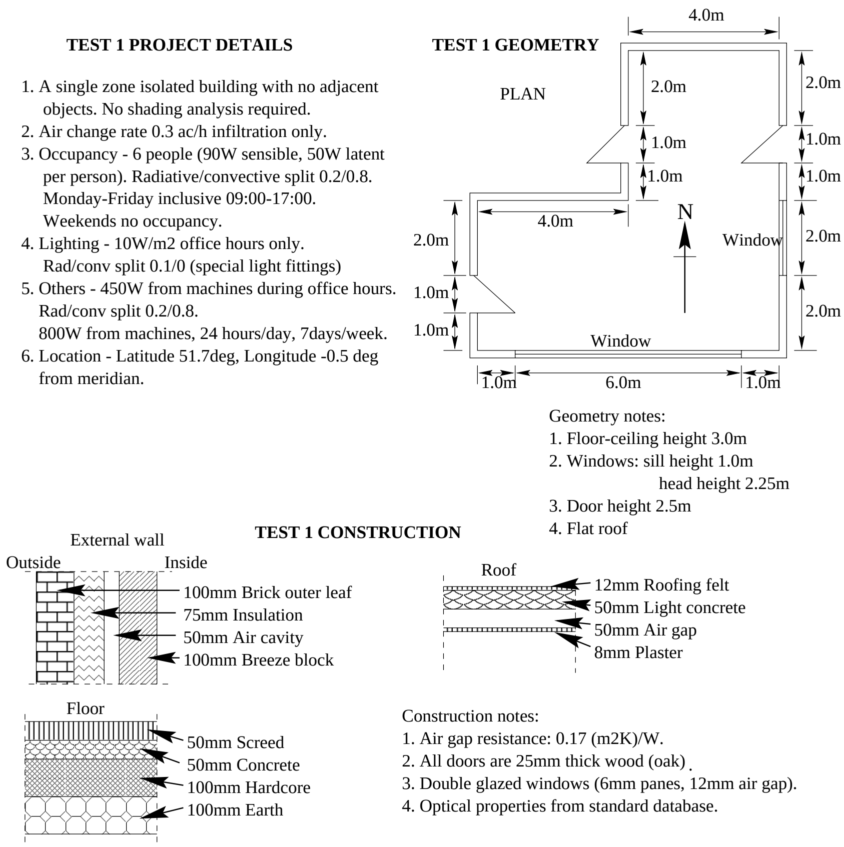


Figure 3.11 The novice's first problem.

especially true for those who have few opinions as to the thermophysical nature of designs. Chapter 8 includes a discussion of skills acquisition approaches for novices.

3.6 Generic Issues

Sections 3.6.1 to 3.6.5 deal with issues which transcend the specifics of project type. These issues appear to be generic and applicable to building performance assessment in general.

3.6.1 User types

Reviewing the participants in the courses and workshops, there are definable classes of users whose responses to simulation tend to follow patterns. The observations do not support a model of a single continuum of users from novice to expert but rather of the following general types.

Previous users of simulation: Those who are already familiar with one simulation tool are, for the most part, quick to adapt to the syntax and philosophy of another, and are receptive to ideas on how best to approach simulation tasks. Almost without exception, such users accept the increased information and interface demands needed to support the additional functionality.

Users with a background in building physics: Those who are familiar with building physics generally have a reasonable aptitude for learning advanced simulation techniques. Typically, their background incorporates a well-defined conceptualisation of building energy flow. Such users tend to be task-oriented and to gain confidence first with their core focus (such as radiation heat transfer or daylighting) before exploring other aspects of simulation. They ask first how phenomena are represented as processes/algorithms as opposed to how they are described in the file store or interface. This is analogous to constructing a mental model of the solution method and then learning how to drive the system. This is a key observation and is used in a subsequent chapter to inform the design of training regimes.

Tool-led users: Users who easily absorb the techniques of how to manage data input and commission simulations, but are not attracted to higher simulation issues (such as methods and abstraction), constitute a class which, for want of a better word, is labelled "tool-led". Others might prefer the label *technician* for those (from many backgrounds) who are mechanistic rather than methodical. Having little or no understanding of the essence of the design, but being nimble of finger, such a user is inclined to enter details until the descriptive limits of the system are encountered, confident that the problem is well represented. With no concept that a subset of a program's facilities might be sufficient, they will inevitably strive to make sure that every possible toggle is set to "ON". The case studies show that the evolution of tools to support the rapid creation of models is problematic for the tool-led. Among many such examples is that of an energetic student in New Zealand who worked on a project related to a university building [Donn 1993]. After two weeks the student posed a number of questions which indicated a model of unsupportable complexity. In one sense, the student clearly mastered many of the descriptive facilities (to the point of attempting to represent stairs explicitly) without requiring assistance. Yet, the composition and complexity of the model were clearly inappropriate.

Single issue users: It is altogether reasonable to find that a tool is not appropriate for a particular project, or may need to be employed creatively to achieve the goals of a project. Single issue users judge the worth of a tool by its treatment of one issue (in which they are likely to have some expertise). Single issue users present a challenge to the instructor because they tend to discount other aspects of simulation and resist balanced approaches to the creation of models.

Domain experts: Domain experts differ from single issue users principally in terms of attitude. In a team of simulationists, one can expect that tasks are divided along lines of expertise. A typical example is that of a mechanical engineer focusing on the assessment of environmental control systems and an architect working on facade details. As discussed in Section 3.4, simulation's support for cooperative working and its place as a common assessment tool, is both problematic and holds great promise. This topic is discussed further in Chapter 8.

Traditionalists: These are users, whether building physicists, engineers or architects, who do not easily cope with simulation. Such users prefer to represent the performance of a design as a limited set of numbers (preferably one) rather than as a complex pattern. They tend not to consider incremental appraisal nor to undertake iterative studies. This group is not easily persuaded of the relevance of simulation and is unlikely to help establish it within design practice.

Students: True novices such as students are an extreme in the continuum of knowledge and opinion/ intuition driving the simulation process. They have little or no domain knowledge, their interactions tend not to be goal directed, they are not well equipped to define performance metrics and do not have the pattern matching skills needed to associate the output of tables and graphs with observed phenomena. They tend not to have matured their observational skills to conjecture a physical equivalent to what is presented on the screen. For example, novices believe that a zone enclosure which appears visually correct in a wire-frame view is also correct in terms of orientation and contiguity.

This class of user is both a bane and a blessing. Being truly naive they attempt the impossible, take dialogue literally, happily ignore pages of warnings (if it was really wrong it would crash) and believe simulation results are carved on stone tablets. Having a capacity to explore they inevitably discover logic gaps. They are worth their weight in gold for the robustness they can force on an application.

3.6.2 Attaining expertise

How novices become simulationists, how professionals can evolve methodical approaches to complex assessments, and how simulation tools can support the acquisition of such skills are important topics in the current research. Yet few seem to have undertaken research into how those with various levels of knowledge in building physics, environmental engineering or professional practice acquire the skills needed to effectively use simulation to support the design process [Hand and Hensen 1995, Hand and Crawley 1997]. The case studies allowed testing of the extent to which simulation tasks might be treated as mechanistic tasks, whether design questions could be anticipated and the skills required to translate designs into models.

Training can and should provide users with an appreciation of the value of reasonable planning and abstraction, even though tool developers have expended considerable resources to support greater model complexity. The case studies indicate that it is good working practice rather than raw power that will deliver useful information. The question is whether alternative tool design and approaches to skills acquisition can encourage this.

Unfortunately, much of the working knowledge associated with the use of simulation is syntactic and is particularly difficult to acquire and retain. In the case of ESP-r circa 1990, a terse interface, "bottom-up" descriptive regime and the user’s responsibility for maintaining model contiguity resulted in a considerable learning curve. In the case studies this often resulted in little time, resource or attention being left to deal with the methodological issues which are essential to the delivery of design decision support.

In the case studies, when simulation-based projects lost momentum, or failed to deliver results, one could inevitably trace the causal factors back to inadequacies in training or an imbalance between technical and methodological skills. The evidence suggests:

- skills acquisition is a critical issue and software can do little to mitigate the inherent complexity of the task of supporting the design process;
- there is a link between the efficacy of simulation-based design decision support and the user’s non-keyboard skills—forming models, choosing appropriate boundary conditions, setting up simulations and interpreting their results;
- mechanistic approaches are ill-advised, and perceptive analysis is not always amenable to automation;
- given the same resources, some teams will arrive at insights to design problems in a fraction of the time taken by others;
- those who acquire keyboard and navigation skills and can create a syntactically correct model may have taken on some of the aspects of an expert without the critical skills needed to "drive the tool";
- students’ rate of progress is generally slower than that of workshop participants, even when they share the same tutorials and are asked to complete the same exercises;
- *foundation skills* are necessary before courses and workshops are undertaken (explored in [Hand and Crawley 1997], Section 3.5.3 and Section 8.1).

The case studies indicated that practitioners could not be assumed to be *au fait* with successful working practices and quality assurance procedures related to simulation. Thus, there are skills and knowledge related to simulation practice which must be explicitly introduced into "best practice" work. Section 8.3 describes several approaches for training within professional practice.

3.6.3 Clues from those with prior experience

One of the more illuminating discoveries has been how those who have a background in building physics or systems design and have experience with one tool are able to use a limited interaction with a mentor to achieve proficiency not normally associated with a self-teaching regime. This is the process they normally use.

- They begin by asking questions which allow them to confirm how particular processes are treated and how facets of a design are represented.
- They tend to discount syntactic knowledge and the particular form of interaction (unless it gets in their way).

- They undertake projects via "what do I want to model?" not " what does the program expect?".
- They are sceptical of predictions—their initial models are usually created to confirm the response of the tool to particular assessments.
- They do not rely on a single simulation but on a series to test the sensitivity of design features and to understand performance.

Experienced users spend considerable time planning their simulations, starting with a sketch of the geometry of their models (and likely future variants) in order to identify where detail is required, where parametric variations may be applied and where/when systems and flow networks will be attached. Experienced users know that issues of calibration, quality assurance and uncertainty are not easily dismissed. These observations provide clues to the additional skills and knowledge—previously unspecified—that would be useful to instructors.

3.6.4 Management expertise

> *methodology 1. a set or system of methods, principles, and rules for regulating a given discipline, as in the arts or sciences. 2. Logic. the study of the principles underlying the organisation of the various sciences and the conduct of scientific inquiry [Random House Dictionary of the English Language, unabridged 1970]*

Managers of a simulation project must be in a position to specify and deploy appropriate simulation tools and staff resources. It follows that the design of simulation should support managers who wish to enquire directly into the progress of a project, assess the work of their staff and have the option to take an active part in assessments.

Consider the translation of design questions, such as "does this building require air-conditioning?" into modelling tasks. If the manager says "find the peak cooling demand", will critical information such as the duration of peak demands (8 hours in the cooling season) or its timing (25°C at 19h45) be included in the project deliverables? An alternative expression of the modelling task is "what will be the peak summertime temperatures and their frequency of occurrence with a naturally ventilated scheme?". This includes the metrics by which a decision can be taken (i.e. number of overheating hours), and the evidence needed to help select a system and the viability of an alternative design. Clearly, a checklist, such as Figure 3.12, is required to focus the practitioner’s attention.

- □ What do you wish to know?
- □ At what level of detail?
- □ What metric signifies acceptable performance and how can this be communicated to the design team or client?
- □ What boundary conditions, over what period(s) would form a reasonable test?
- □ What is the essence of the design in terms of form, composition, operation and control? How might these evolve during the design?
- □ What are the essential interactions which need to be preserved within the model and what performance indices do I need to recover?
- □ How can these interactions be represented? Can I sketch out the overall model and explain my approach to the design team (and if not why not)?
- □ How do I maintain model contiguity, promote quality assurance and ensure that the model can be understood by my colleagues?
- □ How do I calibrate the model and what performance indices do I need to look at to come to an initial understanding of the design?

Figure 3.12 Planning checklist.

It helps if the group has evolved specific procedures, such as the last three steps in the Clarke [1989] methodology which are abstracted below:

- A preliminary report which characterises the energy signature of the design. Overall energy demands and energy balance data are included as required to indicate the efficacy of the proposed design features.
- At this stage the analysis brief may change. Perhaps a number of design modifications are established, with the involvement of the design team, in an attempt to modify energy flow paths. Undirected parametric changes are not favoured.
- The final report is drafted to end the analysis. This may give rise to a second round analysis to focus in more detail on some important issue. For this reason the model is usually composed at a higher resolution than is indicated by the initial brief and the second round can proceed with minimal modifications (e.g. substituting explicit air flow analysis for air change estimates).

The case studies indicated that there is risk associated with detached management (as in "find me the peak cooling demand"), especially if such instructions are approached mechanistically. Teams weak in methodology inevitably compose resource intensive approaches to simulation tasks. Those with poor project management skills are plagued by the chaos that surrounds their work. Those lacking interpretation skills use simulation as a source of formatted summary tables rather than a vehicle for exploring issues that deliver value to their clients. It is essential for simulation processes to be better managed than has been the norm. The question is—how can the design of simulation evolve to support better management of design support activities? This is addressed in Chapter 5.

3.6.5 Abstraction—linking design and model

Simulation acts on an *abstract description* of a design, via a syntax that expresses the underlying data model of the tool (see Section 2.1). The process of arriving at such descriptions is abstraction. Goncalves [1993] discusses the high level attention demanded of the practitioner when exploring trade-offs in model detail. The creation of concise simulation models is an art. Models which are unfocused, simplistic or needlessly complex point to deficiencies in methods and abstraction. In several case studies, attempts at whole building representations failed because the ensuing complexity absorbed resources at an unsustainable rate. At the other extreme, a shoe box representation may ignore critical aspects of a design and lead to inappropriate design decisions.

Abstraction proved a difficult concept in each of the skills acquisition case studies with the exception of mentor based instruction. How can a wealth of options be conveyed to the user? Sketches and screen images can convey useful techniques, but require considerable space in a users' manual. Even if space were not a problem, manuals become obsolete and some aspects of model composition and the relationships within them never seem to find their way into manuals. Other options are needed. These are addressed in Chapter 8.

3.7 Summary

> *" ...it seems building energy analysis is in this [stagnant] condition partly because the Architectural - Engineering - Construction community lacked the vision to foresee it, lacked the will to face it undauntedly, and lacked the teamwork to organize an effective campaign to change it."*
> *[Beranek and Lawrie 1989].*

The sequence of case studies has shown the design process to be complex, ad hoc, iterative, and reliant on interactions within the design team. It is through the testing of design options against a range of performance criteria that designs evolve (tending from the general to the specific, from rough approximations and loose relationships to highly resolved and tightly coupled entities). These observations provide criteria by which to judge the ability of simulation tools to support the design process. Consider the following scenarios:

- Students in a fourth year mechanical engineering class, who have been progressing through a series of set-piece simulation exercises of increasing complexity, suddenly appear bereft of skills when asked to devise and carry out an assessment of a portion of building with which they are familiar. *Interpretation:* The students have progressed to the point where they can recognise some elements associated with simulation but not the relationships necessary to compose a problem.
- A novice, after studying a number of renovation projects, is given access to a CAD package and a set of plans to a similar project and asked to define the simulation model. The result is a literal transcription of the plans which includes, among other things, forty essentially identical offices.

Initial definitions proceed speedily but then stall in the complexity of attribution.

Interpretation: This is a typical mismatch of a technically proficient user and a fuzzy definition of what was required in the project and how to deal with abstraction. The user used one tool to define inappropriate complexity to another tool.

- A team of architects and researchers, who have had a three day introductory course, work in isolation on a complex project. It transpires that model creation has absorbed most of their project resources and fails to support required assessments.

  Interpretation: The team certainly confused keyboard proficiency with the level of decision making required to undertake complex work. Once "over their heads" they kept on rather than seeking the assistance which could have advised them on a more focused model.

- A researcher who specialises in passive solar systems in educational buildings is asked to sketch the thermal relationships and discretisation of a proposed classroom wing prior to commencing work on a simulation model. The resulting sketch contains two rectangular boxes with lumped glazing and mass and no overhangs.

  Interpretation: The constrained descriptions of simplified methods are not in keeping with tools that support explicit models. The researcher should review data model and reporting facilities to identify possibilities for taking advantage of higher resolution.

- In a design study, teams appear to work intensively, yet deliverables are falling short of expectations. On closer inspection, the intense work proves to be a mechanistic adherence to a set of guidelines and reporting conventions: annual simulations being used where a design day assessment would suffice; a whole building approach taken when a focused model would suffice (and at the expense of fenistration and massing detail).

  Interpretation: The team is unable to alter procedures in order to increase deliverables. Management is isolated and not aware that procedures need to be revised.

Each of these scenarios is the result of misunderstandings of the nature of simulation, confusion as to what constitutes an appropriate simulation model and/ or a tendency towards extemporaneous model creation. They are largely independent of the tool and are symptomatic of users who are not sufficiently in control of the simulation process. From the considerable variation in deliverables achieved for projects which were similar in scale and resource, it is clear that how one chooses to use simulation has much to do with its efficacy.

The use of simulation should be driven by methodological approaches rather than by the tool dictating how one approaches simulation tasks. One conjecture was that users would give more attention to simulation methods and appropriate levels of descriptive detail if pragmatic issues of maintaining and evolving models required less attention. Simplified tools such as EDT [Hand 1986, 1987] and Energy-10 [Balcolm 1997], have presented performance patterns relative to a previous or reference design variant. In simulation, the use of base case/ reference case approaches is at the user's discretion rather than a default.

In a set of guidelines for a public utility based on observations of commercial building modelling, Kaplan [1992] suggests that the lack of project documentation promotes distrust of simulation results and many predictions of savings do not occur in reality because the specific conditions (setpoints, occupancy loads, etc.) are not preserved during the specification and construction phases. Kaplan [1992] also observes:

> *Modellers typically turn most of their attention towards the as-designed model. But we have found the baseline model [without energy conservation measures] to be one of the most significant sources of discrepancies between the savings estimates of different modellers. ...[Modellers must] start with an idea of the reasonable range of outcomes.*

In the next chapter these observations will be used as the basis for a specification of a simulation environment which addresses the needs of the building design professions.

References

Balcolm J.D. "Energy-10: A Design-tool Computer Program", *Proc. Building Simulation '97,* IBPSA, Prague, pp49-56, September 1997.

Beranek D., Lawrie L., "Promising (and not so Promising) Developments in Energy Analysis Software" *Proceedings IBPSA BS '89* Vancouver, BC, pp5-10, August 1989.

BDP Energy and Environment, *Planning Report: UK Department of Energy Passive Solar Design Studies, Non Domestic Buildings, Second Phase: Atria and Conservatories*, Building Design Partnership, Manchester, UK, June 1989.

Clarke, J.A. Interview, Glasgow, 1989.

Clarke, J.A. "Assessing Building Performance by Simulation", *Building and Environment*, Pergamon Press Ltd. Vol 28, No. 4, pp419-427, 1993.

Clarke J., Hand J., Johnstone C. "A Summary of Energy Assessments for the Solar Quarter in Regensburg, Germany", *Proceedings Contractors Meeting*, Barcelona, 1996.

Clarke J., Hand J., Hensen JLM., Johnsen K., Wittchen K,. Madsen C. and Compagnon R. "Integrated performance appraisal of Daylight-Europe case study buildings", *Proc. 4th European Conference on Solar Energy in Architecture and Urban Planning*, Berlin, H.S. Stephens and Associates, Bedford, U.K., pp370-373, 1996.

Clarke J., Janak M., Ruyssevelt P., "Assessing the Overall Performance of Advanced Glazing Systems", to be published in *Solar Energy* 1998.

Crawley D. Personal Communication, Glasgow 1997 and Washington D.C. 1998.

Crawley D., Lawrie L., "Beyond BLAST and DOE-2: EnergyPlus, a New-Generation Energy Simulation Program",*Proceedings ACEEE Summer Study 1998*. 1998.

Donn M.R., Personal communication, Adelaide 1993 and Madison Wisconsin, 1995.

Donn M.R., "A Survey of Users of Thermal Simulation Programs" Proc. BS ’97, Prague, Vol III pp65-72, September 8-10 1997.

ESRU, Hypertext training pages: <http:/www.strath.ac.uk/Departments/ESRU]/> 1998.

GA Group, "The Graham Hills Building Feasibility Study", *Report to the University of Strathclyde*, Glasgow, 1993.

Goncalves H. "Building Simulation in Practice, The Portuguese Experience", *Proc. BS ’93*, IBPSA, Adelaide, Australia, pp119-126, 16-18 August 1993.

Hand J. "Thermal Optimisation Techniques for Traditional, Conventional and Passive Solar Building Designs Using EDT", *Proc. Renewable Energy Potential in Southern Africa*, Cape Town, pp16.1-16.17 September 1986.

Hand, J., "The Application of Building Performance Assessment Tools in Professional Practice", *Proceedings of the IBPSA Building Simulation ’91*, Nice, France, August 1991.

Hand, J. and Clarke J.A. "Passive Solar Programme: ESP Support for the Performance Assessment Service, Report for Project E /5A /CON /1249 /2024 Energy Technology Support Unit." ESRU Publication. Faculty of Engineering, University of Strathclyde, Glasgow, November 1992.

Hand, J., Assem E., Strachan P., "Approaches and Constraints in the Simulation of Solar Systems", *Proceedings of the IBPSA Building Simulation ’93*, Adelaide, Australia, pp393-399, August 1993.

Hand, J., Hensen J.L.M. "Recent Experiences and Development in the Training of Simulationists", Proceedings of the IBPSA Building Simulation ’95, Madison, Wisconsin, pp346-353, August 1995.

Hand J, Crawley D., "Forget the Tool When Training New Simulation Users" Proc. BS ’97, Prague, Vol II pp39-45, September 8-10 1997.

Hand J., Crawley D. "Forget the Tool in Simulation Training", *Building Performance*, Building Environmental Performance Analysis Club (BEPAC), Issue 1, pp3-5, spring 1998,

Kaplan, M. B., Caner P., Vincent G. W. "Guidelines for Energy Simulation of Commercial Buildings", *Proc. ACEEE Summer Study*, ACEEE, 1992 Vol 1, pp137-147, 1992.

Rittelmann P. R., "Achieving Architectural and Engineering Collaboration", *Proc. Total Building Design Converence*, ASHRAE, Chicago, pp89-107, October 16-17, 1995.

Selkowitz S., Haberl J., Claridge D, "Future Directions: Building Technologies and Design Tools",*Proceedings ACEEE Summer Study 1992. Vol 1, pp269-290, 1992.*

Sluce A. *Passive Solar Design Studies of Non-Domestic Buildings: Phase II: Atrium Design Studies Design Reports*, ETSU S/N1/00081/REP/B Building Design Partnership, Manchester, 1993.

WHAT IS REQUIRED

4 WHAT IS REQUIRED

It is instructive that experts rarely approach the physical limits of a simulation environment. Others are less wise. (Statement missing from the preface of users' manuals.)

Despite the considerable benefits which can be derived from the use of simulation for design decision support, the case studies have identified constraints vis-a-vis the evolving nature of assessment questions during the design process, the complexity of the built environment, gaps in quality assurance tasks and the ability of simulation to work cooperatively with other design tools. This chapter takes the issues identified in the case studies with the observations of others (from Section 1.1) to define aspects of a simulation environment more in keeping with the needs of the design process.

Of the many surveys of the developer and user communities, the series of developer and user workshops by Lawrie and Crawley [1997] is particularly instructive. The consensus which emerged from the developer community included the need to develop tool interfaces which would adapt to the stage of the design process (such concepts having been demonstrated in the IFe project [Clarke and Mac Randal 1991]).

At a more pragmatic level, developers believed the essential ingredients of the next generation of tools to be intelligent defaults, extensive component libraries, object-oriented representations, interoperability with other tools, context-sensitive help, and the ability to customise inputs and reporting.

The user community wish-list was that next generation simulation tools would facilitate collaborative and integrated building design, support early analysis of design alternatives, promote education of students and practitioners, and facilitate the assessment of comfort and environmental issues via flexible descriptive facilities and reports.

The current work draws on such earlier work, but with some scepticism. Why? Too often, tool evolution has dealt with symptoms and tool vendors have been unwilling to expand the boundaries of simulation. Worse, many in the simulation community maintain an attitude that simulation work begins with the definition of a model and ends with the production of a set of predictions. In Europe, especially within the University of Strathclyde, the revision of the boundaries of simulation (e.g. into network and CFD air flow, explicit power modelling) and of its use by practitioners continues apace.

4.1 Review of observations and issues arising

The principal observations and issues from the case studies are summarised below. The design studies revealed that simulation should:

- be open to intermittent use, without the need for prior knowledge of simulation models, and should present information concisely, unambiguously and in terms which are familiar to the user;
- accommodate the considerable compositional and operational complexity observed in the built environment;
- allow the focus of assessments to evolve over time—e.g. from system capacity assessment to glare discomfort evaluation.

The consulting case studies revealed that:

- simulation should evolve to better tell *the story* of the design to the simulationists and design team;
- simulation practice should evolve to create models which are less opaque to others in the design process;
- work practices and tool facilities which hinder the rapid evolution of models are problematic;
- audit trails (e.g. decision points, assumptions) warrant consideration as part of the data model;
- quality assurance is central to consulting, and the resource needed to undertake this task is almost always underestimated.

In terms of skills acquisition, the case studies revealed that:

- deficiencies in training are expressed in terms of poor quality assurance, chaotic and complex models, missed deadlines and frustration;
- those with prior experience in simulation start by matching their mental model with that of the tool before confronting the new tool's syntax;
- the breadth and depth of options can be visually overwhelming to the novice;
- hard copy instructional information quickly becomes obsolete, is unsuited to methodological topics, is unwieldy as a reference for scores of user dialogues, but is easier to scan than text on the screen;
- example problems may illustrate syntax but they do little to show the user how to go about solving difficult (i.e. real) problems;
- training within the context of an ongoing project exposes participants to the *process* of simulation (decisions, interactions, assumptions, etc.) so that they emerge with considerable capabilities.

Overall this suggests the need for simulation to adapt in ways which will allow assessments to be carried out within the time and resource constraints of the design process.

4.2 Tool requirements

To guide the discussion of possible approaches to the issues arising from the case studies, a number of broad goals are formulated and then the specific mechanisms for implementing these are presented.

Shifting the point of nominal model complexity

One of the constraints of simulation is the degree of complexity that can be accommodated within the limited resources of the design process. Such complexity is seen both in the interplay between thermophysical processes and within the design process itself. If a simulation model is viewed as a matrix of descriptive entities, the number of possible interconnections grows quickly as the number of entities increases. While each entity may require only a moment to define, it must be attributed, linked into the model, syntax checked and, eventually, performance checked. Thus, each tool has a point of *nominal model complexity* which requires a modest investment of time and attention by experienced users. Beyond this point, additional complexity becomes increasingly costly in terms of user attention and quality assurance demands.

The point of nominal model complexity appears to depend on the facilities offered by the simulation tool and the demands placed on the user by model creation. For example, in the DEROB thermal simulation suite, this point was initially reached at about fifty surfaces, but a revised interface shifted this to several hundred surfaces.

Many users are preoccupied with the task of creating model geometry and thus it is fashionable to envisage that a close integration with CAD tools will solve many of the perceived problems of simulation. It is less fashionable to point out that users tend to underestimate the resource required to deal with the attributions (thermophysical, operational, topological and control) associated with simulation or that most CAD tools lack attribution facilities. One must address the task of attribution (and the underlying contiguity of the data model) as much as the visible joining of vertices into surfaces.

In the case studies, those who attempted to create and maintain whole building models often found themselves beyond ESP-r's point of nominal complexity. In order to meet the demands of the design process, it is necessary to shift this point to be more in keeping with the complexity observed in typical projects.

Enabling cooperative working

It is increasingly the case that design teams rely on a number of tools and facilities for sharing information. In the decade since Sonderegger [1989] called for simulation to take advantage of commercial databases, graphing tools and text editors some progress has been made. There is a case to be made for simulation not to duplicate the functionality of such tools. At the same time, it is reasonable to enquire whether simulation is always well served by entrusting descriptive and analytical tasks to external tools, as is the case with DOE-2 and its proposed successor Energy Plus. If there are lucani in support for the creation, documentation, proving and evolution of models, there are quality assurance implications. If access to performance results is dependent on external filters, causal relationships may be overlooked, lost or distorted.

Unfortunately, the isolated development of simulation tools and the fiendishly difficult task of exchanging fundamentally different data models (see Section 2.3), or of passing data though a restrictive medium (e.g. text files, message passing convention), has hindered the cooperative use of tools. An example of this is the "market leader" DXF format, which is distinctive in its age and verbosity. Table 4.1 illustrates a few of the data model differences which make it difficult to reliably exchange information with CAD applications.

Table 4.1 Constraints in exchange media.

Entity	ESP-r	DXF
Zones	space fully bounded by polygons	no such concept or mode of expression
Surfaces	limited number of polygons with 3-24 edges	thousands of polygons with 3-4 edges
Glazing	polygon, optical and thermal attributes	usually implied by absence of surface
Obstructions	rectangular block	block
Surface topology	enforced	user decides
Boundary conditions	one per face of known types	no such concept
Attribution	name, composition, boundaries	blocks and layers can have names
Layers	none	named, colour and line type attributes
3D grids	none	extension to 3D surfaces
2D entities	none	lines, faces solids, extrusions
3D entities	surfaces, blocks, ground topology	3D lines, solids, extrusions, 3D polygons
Rotation entities	approximated by sets of surfaces	automatically generated

From the standpoint of simulation, the (relative) lack of attribution and contiguity, and glazing systems represented as the *absence* of a surface within CAD models is problematic. Typical responses to the user being confronted by 450 unnamed polygons (150 of which are reversed), are to:

- use arbitrary conventions (such as placing each simulation zone or material on a separate layer);
- create a software harness to coerce the CAD application to provide information needed;
- adapt simulation via expanded internal CAD facilities and allow a user to revise (e.g. attribute, transform, rotate, mirror) imported models to the requirements of simulation.

Such pragmatic approaches do not address the core problem of the lack of a common data store within a project and a practitioner’s inability to orchestrate the use of a palette of tools. This is discussed in Chapter 7.

Redirecting the user's attention

The objective of the original ESP-r input management program was to create and evolve files. Dialogue with the user was optimised for the piece-wise alteration of model details at the expense of being able to enquire about model features or record the rationale for a particular approach. What is required is to evolve not files but models. In the detailed design case study, users were so caught up in the mechanics of interaction—recording information on scraps of paper, trying to reconcile previous changes in the model and thumbing through the users' manual—that they were often distracted from the goals of the project. Such complaints could be directed at all building simulation tools.

Tool design can mitigate many distractions. The tool can, for example, provide facilities to copy and edit entities rather than force the user to record details, it can retain information (such as file names), make useful suggestions for data to be entered, and provide a project log to record assumptions and approaches. The attention given to interaction with the tool is, to some degree, a function of trust. Users gain trust by repeated successful use, eventually reaching a point of competence. Inconsistent treatment of user actions, lack of feedback on the implications of user actions and the inability to escape from inappropriate selections act to reduce trust.

Viewed from this perspective, a hierarchical, but static menu selection supports navigation without informing the user of the current state of the problem. It is useful to consider what information might help the user make more informed choices. For instance, in scanning through thermal zones certain details (name, volume, overall areas, whether attribution is complete, etc.) not only reduce ambiguity but moderate short term memory demands (e.g. questions like "which entity am I editing" after glancing away from the screen).

With the exception of planning sketches, if a user is forced to record information on paper this not only interrupts the descriptive process but is indicative of inappropriate or inadequate feedback and possibly a gap in the documentation included in the data model. An example is the process of inserting additional vertices in a polygon (perhaps as a prelude to subdividing the surface). Clearly, writing down sets of numbers or having to use a calculator is less direct and more error-prone than a "copy & edit" or in-built trigonometric function.

Supporting "what ifs"

"What if" questions are central to the design process and imply the need to focus rapidly on a particular aspect of the design. Usually, this is accomplished by altering the level of detail (granularity), composition or operational nature of the model. For example, if the user wishes to appraise the sensitivity of a thermostat to periodic direct solar radiation, then the simulation program should allow an explicit representation of the sensor and the room in which it is placed, as well as support assessments at a frequency related to the response time of the sensor.

Some simulation tools provide discrete levels of granularity while others provide a general syntax. An example of the former is Energy-10 [Balcolm 1997] which assumes that a 1000 m^2 design can be represented by a pair of zones and that the effects of internal mass, radiant exchange, temporal solar radiation patterns and geometric complexity can be treated in a simple manner. At the other extreme there is a general syntax which allows explicit modelling of objects, such as the aforementioned thermostat.

Energy-10 imposes a granularity, but in doing so, makes it possible to cycle quickly through a number of designs. The cost of this is that the range of "what if" questions is constrained. The latter general syntax does not impose granularity and thus has the potential to focus more closely on the needs of the design assessment—assuming the user understands the nature of the problem and the facilities offered by the tool. The cost of the latter is the need to master a descriptive process which is both minimally prescriptive (in terms of task ordering and compositional detail) and able to resolve dependencies.

What if questions also required that the practitioner confirm that the resulting energy signature matches expectations. The term *energy signature* is intentional as it implies recognition of a pattern rather than a cursory inspection of maximum and minimum temperatures. For example, lightweight and massive buildings have recognisable responses to periodic or step changes in boundary conditions, PID controllers are associated with a different pattern of actuation than an ON/OFF controller and the closing of blinds to control glare is often linked to an increase in casual gains from lighting fixtures. Clearly, the diverse assessment demands of the design process are more likely to be supported if simulation offers multiple views of performance at different levels of detail. An example of this is an ability to investigate the specific implications of a design change or to track the causal elements of poor performance. An extension of energy balance reports in the results recovery module to support such "causal chaining of energy balances" [Clarke 1993] is shown in Figure 4.1.

The need to support "what-ifs" extends to the reporting facilities built into the software and the formal reporting which is often the primary delivery mechanism to other members of the design team and client. Simulation needs to deal with a number of constraints in the analysis and reporting processes.

- A report must prejudge what is to be presented. The resource needed to cover a range of issues at several levels of detail is considerable.
- Presenting sequences of complex relationships, such as insolation and shading patterns, can be awkward.
- Interactive use carries the risk of the tool crashing.
- Simulation tends to be optimised for domain experts (who may question the ability of others to deal with domain specific data).
- Managers know how to plan meetings around reports, not interactive tool use.

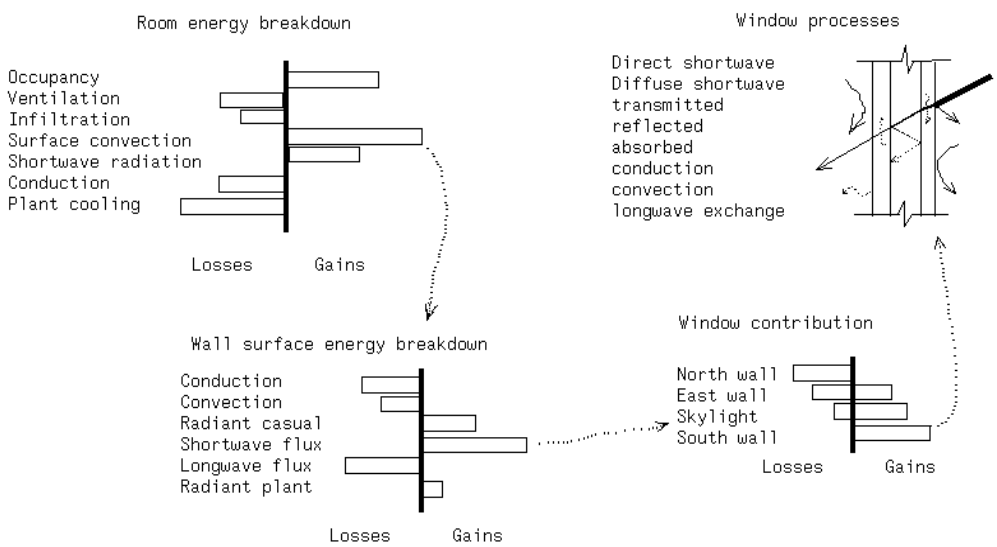


Figure 4.1 Causal chaining of energy balances.

What if design meetings were typified by interactive explorations rather than a discussion of reports? To the extent that simulation can support design questions interactively, any other approach (such as a formal report) must introduce a time penalty and, in practice, constrain the information available to the design process.

Persistent models

The case studies showed that simulation models often had limited persistence. Simulation teams had difficulty replicating studies which were more than a few months old. The resources required to understand a project acquired from another source or drawn from an archive were found to be a considerable deterrent to their use. Simulation teams working on several projects found it difficult to shift attention between projects. Technology transfer initiatives were effectively precluded from distributing simulation models as project deliverables. It also proved difficult to locate models for use in workshops and formal courses on simulation.

These shortcomings constitute a barrier to the deployment of simulation which is no less serious than the difficulties associated with learning the effective use of simulation or difficulties in defining models. Important contributors to this limited persistence of models have been identified as:

- poor documentation of models by users (e.g. meaningless labels, failure to use descriptive facilities), encouraged by limitations in the data model (e.g. no mechanism for recording notes or observations, limited storage allocations for descriptions);
- a data store optimised for machine reading rather than for the needs of archivists or to support visual scanning;

- a flat file store (e.g. data partitioned by file but not by a hierarchy of folders in the file store);
- few guidelines for setting up and managing projects so that persistent models are considered part of best practice.

The option to be less cryptic involves removing the interaction penalty associated with documentation, extending the data model to accept additional documentation and providing early evidence that shortcuts are counterproductive.

Rapid assimilation of models

Practitioners are often required to shift attention between projects. Clearly there are limits to a user's ability to maintain a coherent mental model of multiple projects, and after a project has been idle for several months, there is a need to "come back to speed". Rapid assimilation requires the recipients to become *au fait* with the nature and intent of the model and the assumptions that have been made in its generation.

Unfortunately, simulation has made few concessions to those who wish to understand another group's model or to those who wish to share their models with colleagues. Reducing dependence on prior knowledge and the opacity of models are issues that must be urgently addressed.

The technique used in the current work involved recording occurrences of users having to jot down information during the life of consulting projects and during training workshops. Such pauses invariably break the flow of the task—such information should be persistent. Another set of observations recorded where and how often information was transformed from one format to another. It has also been useful to consider the practitioner's view of "what would make it difficult to recover an archived project and replicate a previous study":

- missing details as to the form of the model (especially if highly abstracted);
- missing details as to the options invoked within a particular simulation;
- missing documentation as to the intent of the prior study, the methodology adopted, reasoning behind the abstraction chosen as well as expected points of uncertainty and assumptions made.

What is required is the facility:

- to record and review project goals, methods and assumptions made;
- to review the current state of the model in order to check its composition and to rebuild a mental picture;
- to confirm the coherence of the model, and the availability and syntactic correctness of files and databases.

Support for ad hoc, incomplete and evolving designs

Consider the pattern: an architect composes a design using particular climate and environmental control assumptions which are checked by passing the (substantially complete) design to an engineer. Such a regime makes it difficult to optimise the whole and tends to delay decisions to a point in a project where shortcomings are not easily remedied. In ESP-r circa 1990, the display and specification of geometric entities, their composition, boundary conditions, etc. were contained in separate facilities which were usually accessed sequentially in terms of geometry, composition, operation, contiguity and control. This delayed the exposure of inconsistencies and proving of models.

The case studies showed that the design process is not well served by a "bottom-up" approach where the model becomes viable only when all the pieces of the puzzle are in place. The design process is also characterised by a considerable variability in supporting data. There must be some provision for incomplete models and for a degree of flexibility in the descriptive process. Simulation should support the rapid and robust *evolution* of models so that the iterative use of simulation becomes unremarkable.

What is required is support for an ad hoc descriptive process where the user's focus of attention is not prescribed. It would also help if the marginal cost of additional assessments or modifications to models was low. Practitioner pleas of "it works, please don't make me change it" must become, "fine, give me five minutes to do a global search and replace a material type and see what difference a low-e coated window makes to comfort and condensation risk...".

It is conjectured that if data model entities were treated as (and viewed by the user as) objects which could be identified, copied, deleted, created and attributed, then one of the major constraints to meeting the demands of the design process would have been addressed.

Clarke [1998] suggests that the ad hoc and incremental nature of the descriptive process should also be reflected in incremental access to assessment functionality. For example entries in construction databases provide "U" values, the addition of zone geometry provides access to information on shading patterns and further attribution should give rise to glare predictions and the identification of condensation risk. Norton and Lo [1991] discuss how simulation might ensure that information from simple assessment tools can be incorporated into tools used in the later stages of design.

Quality assurance

> *Models are to error as sponges are to water [Kaplan 1992].*

In consulting, partly because of legal liability for the effects of error, quality assurance is, in the minds of many practitioners, the *sine qua non*. The need for unambiguous exchange of information within the design process is clear. Simulation presents any number of difficulties in this regard, from the frequency of user data input errors to the misunderstanding of data requirements or of the intent of a modelling exercise. Each of these may lead to misleading or inappropriate feedback to the design process.

In design there are formal modes of communication by way of plans and briefs, which support informal communications. The simulation community has no *common medium of exchange* and is thus burdened by jargon, and competing modes of expression. For example, zone temperatures in SERI-RES are a weighting of air and surface temperatures which are not directly comparable with zone temperatures used in other tools.

In ESP-r, a wall or ceiling is a *composite construction*, while aluminium is a *material*. In displays and reports units are sometimes vague, indices are used rather than descriptive expressions, and data model terms are used rather than those of physical objects. These combine to form a jargon and a syntactic barrier. Careful design of interactions is required to improve the context in which model details are presented.

The number of data items required to define simulation models of moderate complexity causes considerable unease for those concerned with quality assurance issues [Parand and Bloomfield 1991]. Some researchers [Chapman 1991] argue that data input errors increase linearly with the number of items (with an average of 7 errors per 125 data items), while computational error tends to decrease with the logarithm of the number of data items (i.e. moving from abstract to explicit representations)—and so the overall error tends to rise as more information is required. For assessments of existing buildings the classes of errors reported by Chapman are: a) observational errors during the building survey, b) conceptual errors in the abstraction of observed detail, c) inconsistency in information gathering procedures (taking the wrong measurements), d) measurement and scaling errors and e) keyboard errors.

The rate of keyboard errors reported was in the context of dimensional and numerical input into a simplified design tool where no graphical feedback was provided. The extent to which interface design and internal program checks can mitigate errors is discussed in Sections 4.3, 5.1.2 and 5.1.5.

With a novice, differences between what is intended and what the simulation tool recognises are expected. In the context of professional practice this is catastrophic. The case studies identified that:

- loss of model contiguity during composition and editing reduced productivity;
- distributed information sources hindered the understanding of models;
- reporting facilities could be factual without being informative;
- bottom-up descriptive approaches delayed contiguity checks;
- users were fallible and increasingly inefficient and error prone as the complexity of a problem grew.

Consider the above mentioned loss of contiguity (e.g. adjacencies and boundary conditions at each surface). It is computationally efficient to derive a list of possible geometric associations, but where more than one association is possible (two rooms separated by a zone that explicitly represents a cavity wall) the user should be the final arbiter. Thus the tool should perform useful modifications to a model, helping to reduce the tedium associated with simulation tasks without reducing the user's awareness of the implications of such instructions.

Another historic shortcoming of simulation is the limited persistence of models and the general inability of simulation teams to reuse, let alone replicate past studies. Quality assurance requires that models evolve from being viewed as a set of machine readable files to become a transportable, coherent, attributed and documented superset of project data.

4.3 Mechanisms of change

Having identified broad goals in terms of redirecting the user's attention, rapid assimilation of models and evolving designs, the task is to identify aspects of simulation's data model, functionality and simulation practice which must evolve. The discussion should be viewed as generic to all simulation tools.

Attribution and naming conventions

If simulation does not tell *the story* of the design then how can it support the design process? To name something is to *own it* and the act of naming usually forces one to *observe and understand* the object. Consider the following two descriptors: "surface_4 in zone_8 is composed of the sixth database entry" and "floor in office12b is composed of carpet_on _slab". One makes the user work (to confirm position, composition and contiguity), the other helps tell a story. Such patterns are also evident in the organisation of project files shown in Figure 4.2.

```
omega: ccas05 %ls
 q.001.cfg   q.z03.utl   q.z07.opr   q.z11.geo   q.z13.utl
 q.001.ctl   q.z03.utl1  q.z07.opr1  q.z11.obs   q.z13.utl1
 q.002.cfg   q.z04.cgc   q.z07.utl   q.z11.obs1  q.z13.utll1
 q.003.cfg   q.z04.con   q.z07.utl1  q.z11.opr   q.z14.cgc
 q.003a.air  q.z04.geo   q.z08.cgc   q.z11.opr1  q.z14.con
  . . . . [cut section]
 q.z03.geo   q.z07.cgc   q.z11.cgc   q.z13.geo   q.z17.geo
 q.z03.opr   q.z07.con   q.z11.cgc1  q.z13.opr   q.z17.opr
 q.z03.opr1  q.z07.geo   q.z11.con   q.z13.opr1
omega: ccas05 %

omega: jon %ls
 delta_gl1.cfg   atr_top.utl   east_gr.opr    north_gr.geo   plant_rm.utl
 delta_gl1.ctl   atr_top.utl1  east_gr.opr1   north_gr.obs   plant_rm.utl1
 delta_gl2.cfg   atri_mid.cgc  east_gr.utl    north_gr.obs1  plant_rm.utll1
 delta_gl3.cfg   atri_mid.con  east_gr.utl1   north_gr.opr   south_gr.cgc
 delta_gl3.air   atri_mid.geo  entry.cgc      north_gr.opr1  south_gr.con
  . . . . [cut section]
 atr_top.geo     east_gr.cgc   north_gr.cgc   plant_rm.geo   west_gr.geo
 atr_top.opr     east_gr.con   north_gr.cgc1  plant_rm.opr   west_gr.opr
 atr_top.opr1    east_gr.geo   north_gr.con   plant_rm.opr1
omega: jon %
```

Figure 4.2 Folder naming conventions.

A file naming convention as opaque as the upper portion of Figure 4.2 is usually indicative of terse documentation at all levels of the model. The lower portion of Figure 4.2, remains terse, but at least provides an indication of contents. The importance of naming schemes was confirmed in an earlier study [Hand 1986], where the naming of entities was seen to enhance the understanding of both models and reporting.

Where users take care to annotate their work and assumptions, they also tend to use naming conventions which reduce the barriers to others’ understanding of their work. Observations of consulting projects provided indications of where practitioners enhanced existing documentation. Table 4.2 shows a raw data file and a manually annotated version. Clearly where such information aids understanding it should be incorporated into the data model and, where possible, used in the interface and in reporting.

Table 4.2 User naming/ attribution conventions.

Raw geometry file	**With manual annotations**
-	# van description: front seat (zone 2)
GEN	GEN fr_pass
25, 15, 0.000	25, 15, 0.000 # 25 vert, 15 surfaces, no rotation
0.7350, 1.9650, 0.3000,	0.7350, 1.9650, 0.3000,
1.5000, 1.9650, 0.3000,	1.5000, 1.9650, 0.3000,
1.5000, 3.3550, 0.3000,	1.5000, 3.3550, 0.3000,
. . . .	. . . .
0.7350, 3.3550, 1.1430,	0.7350, 3.3550, 1.1430,
0.7350, 1.9650, 0.8430,	0.7350, 1.9650, 0.8430,
1.5000, 1.9650, 0.8430,	1.5000, 1.9650, 0.8430,
	# surface vertex list
6, 4, 5, 1, 24, 20, 23,	6, 4, 5, 1, 24, 20, 23, # R_side
4, 1, 2, 25, 24,	4, 1, 2, 25, 24, # Firewall
5, 2, 3, 22, 21, 25,	5, 2, 3, 22, 21, 25, # Filler
4, 20, 21, 22, 23,	4, 20, 21, 22, 23, # Base
11, 1, 5, 6, 7, 8, 9, 6, 5, 4, 3, 2,	11, 1, 5, 6, 7, 8, 9, 6, 5, 4, 3, 2, # Seat_R
. . . .	. . . .
4, 13, 12, 15, 14,	4, 13, 12, 15, 14, # Seat_B
	# no windows
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
	# no doors
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
3, 0, 0, 0	3, 0, 0, 0 # diffuse insolation distribution

Exemplars

Both novices and practitioners can benefit from studying models of realistic complexity. Unfortunately, many models which demonstrate useful approaches are poorly documented and ill-structured. Not only does this make them opaque for instructional purposes, such artifacts also hinder cooperative working and the exchange of information within design teams. Clearly, if practitioners are to learn from successful projects it is critical that models evolve into self-contained *exemplars* which can be disseminated and understood by others.

An exemplar should allow others a reasonable expectation to understand and then carry forward a project. Thus, conversion of a model into an exemplar demands clarity in all aspects of the project including the design questions it was intended to support, the level of detail selected, assumptions, operating regimes and boundary conditions.

Data models do not include such documentation and attribution, and the exchange of simulation models is far from a trivial exercise, so the realisation of exemplars, with corresponding enhancements to the data model and simulation facilities is a key deliverable of the current research and is discussed in Chapter 5.

Tutorials

One problem with context-sensitive help is that it may be overly terse for novices who do not understand the terminology used, or for experts who require detailed descriptions. Addressing the needs of such users would add considerable verbosity, so a facility for extended discussions would be helpful.

There are many instructional topics which do not easily fit in manuals or context-sensitive help within a tool. For example:

- explanations of entities in the interface (in an evolving system, a users' manual can quickly become obsolete);
- relationships between modules within a simulation suite and the part they play in the simulation process;
- extended topics—such as the circumstances where shading patterns might be important.

Unsurprisingly, this description resembles that of help facilities attached to commercial software. They have in common a desirability of ad hoc access and juxtaposition with the model on the display, rapid and low cost updating, possibilities for cross referencing and topical searches. This is balanced by the low resolution of the screen (see [Tufte 1990]) slow scanning of such displays (in comparison with the printed page) and the need to learn yet another interface. They can also tax the user's short term memory if they obscure the tool they are meant to support or require the user to toggle between displays.

Clearly, a tutorial facility would address a gap in support (between that of users' manuals and context-sensitive help) and allow the frequency and nature of access to be studied. This required consideration of topical organisation and content, presentation modes and points of access from within the simulation environment. This is explored in Section 5.1.6.

These mechanisms of change (attribution, exemplars and tutorials) will assist in the overall goals defined in Section 4.2, but only if consideration is given to the interface.

4.4 An unnoticed interface

One myth about simulation is that it would be easy if its interface were easy. Certainly it is possible that some aspects of simulation would be less tedious. However, ease-of-use does not change the decisions related to simulation methods, the composition of concise models, the selection of appropriate boundary conditions or the interpretation of predictions. Simulation is primarily a tool to answer serious performance appraisal questions and this thesis is primarily concerned with the introduction and testing of project management facilities. In this context, a reasonable interface goal is to allow users to concentrate less on interacting with the tool and more on using the tool—hence the inclusion of a pragmatic, and hopefully unnoticed, interface which cheerfully sacrifices appearance for the requirement of this brief.

The subject of human-computer interfaces has been dealt with extensively by a number of authors. The work of two authors, Ben Shneiderman, in particular his *Designing the User Interface, Strategies for Effective Human-Computer Interaction* [Shneiderman 1993] and Edward Tufte's definitive series *The Visual Display of Quantitative Information* [Tufte 1983], *Envisioning Information* [Tufte 1990], and *Visual Explanations* [Tufte 1997] support many aspects of the interface design used in the current work.

Those who deal with human-computer interface issues classify tool interaction within a matrix of tasks to be undertaken (and associated risk), speed and frequency of interaction as well as skills brought to the tool. In simulation, the *need to limit risk and ambiguity takes precedence over speed of interaction or accessibility to novices*. Because of the range of tasks and the possibility of intermittent use, critical choices are not likely to be made on the basis of rote response. Because potential users of simulation come from several professions, they are unlikely to share a common vocabulary or set of expectations, so clarity is critical in the design of interactions.

The research bias of most simulation tools is evident in the degree of syntactic knowledge required of the user. Examples from ESP-r are: in the materials database, air is an entity which is not explicitly included, but is represented by the index 0; a construction is defined from the outermost layer inwards. Such knowledge is difficult to acquire and to remember.

A tool's interface hierarchy (number of levels vs. breadth, number of items on each menu, etc.) also has an implication on the user's ability to find facilities and makes considerable demands on short term memory. In the case studies, the acquisition of navigation skills proved time consuming.

In contrast, conceptual knowledge is more likely to be remembered by the user and to inform interactions. For instance, if the concept that a thermal zone is a volume of air at one temperature (i.e. well mixed) and fully bounded by polygons is adopted, then the formulation/ abstraction of the problem can proceed irrespective of any arbitrary convention of the ordering of edges.

The case studies show that two essential elements are required and these are clarity and consistency. Firstly, an interface must have clarity—ambiguity is not in keeping with a professional's need for accuracy, the novice's need to acquire skills or the student's need to understand the temporal patterns of energy flows. Secondly, there are benefits from consistency in interaction, whatever the specific form of the interface.

Interface clarity

What constitutes clarity within a simulation environment? One might start by looking at the tasks of model creation, proving models, enabling assessments and presenting results.

Model creation: To mitigate information demands some tools constrain geometric or operational complexity. What tends to go unnoticed is that this can obscure the nature of the problem and make it difficult to detect errors [Crawley 1995]. For example, standard DOE-2 test cases, when checked with a third party visualisation tool, were found to have surfaces which floated tens of metres from their context. Thus, savings in initial descriptive tasks must be weighed against ambiguity.
At the other extreme, where simulation includes a CAD-like visual context there is a tendency to assume that this obviates the need for naming entities. This ignores other contexts such as reports and entries in lists where names are a valuable aid. It also deprives the user of a potent mechanism of observation and ownership while adding a burden to those who seek to understand the nature of the problem.

Proving models: The tasks associated with proving a model are similar to those of assimilating and understanding a model and have been covered in Section 3.6.4. Typically, the user scans the model and its predicted performance for entities that do not fit, or for unexpected patterns. Multiple views of the model composition and its performance would assist this process.

Enabling specific assessments: Ideally, the creation of a simulation model is based on the rules of abstraction and the methods used in the project and once these are established the balance of the work should be straightforward. However, assessments are based on a collection of descriptive elements, supporting databases and simulation parameters and each of these choices should be explicit in the user interaction and preserved so that assessments can be replicated at some future date.

Present predictions in forms that are understandable to the design team: This is complicated by an inherent conflict between the developer's need to define general facilities and the specificity of project requirements. Thus, a summary of solar radiation entering, leaving and absorbed in a zone is applicable to many projects but a request to conflate several flux paths into a single index is to be discouraged. There is also the need to support both interactive reporting (e.g. tables and graphs) and data to be exported for high quality reporting/ analysis.

To find out what issues were confusing to users, notes were made during workshops and classes as to what portions of the interface were problematic for various user types and levels of expertise. Where possible, the confusing entity was revised during the period of the workshop to see if an alternative approach reduced confusion. What transpired was that removing a source of confusion for one type of user invariably improved the understanding of others (or at least did not reduce the clarity for others). For example:

- Much of the confusion with the definition of polygons as ordered lists of vertices was resolved by including a reminder that the list should be anti-clockwise when viewed from the outside and providing immediate graphic feedback where problems were detected.
- There were cases where a set of unique choices (e.g. which surfaces to distribute incoming solar radiation) were stored in a sequence of variables which were opaque to the user. Replacing this with a list selection and clear reporting clarified the interaction.

Perception of clarity is reduced by the visual chaos of layer upon layer of indistinguishable pop-up boxes and potential points of interaction. Ease of creation of interface objects does not necessarily correlate with the design of interfaces which support the needs of simulation. Careful consideration of the context and content of interaction is required.

Interface consistency

The second critical attribute of an interface is consistency. The case studies indicate that confidence is eroded and productivity is reduced when a tool provides inconsistent modes of interaction, makes repeated demands for the same information or loses information because it was not known to be volatile or subject to an unresolved dependency. Distrust can isolate the user from useful facilities. A case in point is the provision of context-sensitive help. A survey of students showed that where they were disappointed with initial access (e.g. not available, overly terse or technical, badly written), they tended to discount the value of general help facilities and to rely on fellow students, users' manual or the lecturer for support.

Other aspects of consistency have to do with reinforcing learned responses. If all list selections and requests for numbers and text are handled consistently, regardless of the style adopted, then once the user gains that skill it ceases to be an issue. The classic benchmark for this is the Apple Macintosh interface. Having mastered one application, subsequent applications are often taken up with no more than a cursory reference to printed manuals or on-line help.

One of the hallmarks of a productive tool is that it allows the user to focus attention on the work to be done rather than on the means of getting the work done. Where the placement and ordering of interface elements (charts, wire-frame images and the like), text feedback, user dialogue, and command selection is consistent then they also tend to become unnoticed. The frustrations and inability to navigate within the tool which is evident on the first day of a training workshop tend not to be issues by the middle of the second day.

4.5 Summary

Whatever the evolution of the interface, this must be balanced by other aspects of simulation practice. It is necessary to consider the demands of the design process: how firms deploy simulation; how users acquire skills; the nature of interactions with the user and with other tools as well as the design of tools and the facilities they offer.

A first step is the recognition that the design process is **project based** and demands information outwith the traditional definition of simulation models as well as new modes of interaction with the design team.

This gives rise to a new description of simulation:

- which is applicable at various stages of a project;
- a conduit and repository of information within the design process;
- approachable by the manager directing the assessments, the consultant who requires access to detailed performance data or a "what if" tool in a design meeting;
- where the caricature of researchers in white lab coats who belatedly deliver pronouncements as if they were carved on stone tablets was only a memory.

References

Balcolm J.D. "Energy-10: A Design-tool Computer Program", *Proc. Building Simulation '97*, IBPSA, Prague 8-10 September 1997, pp49-56, 1997.

Chapman J, "Data Accuracy and Model Reliability", *Proc. Building Environmental Performance, BEP '91*, BEPAC, Canterbury, pp10-19, April 1991.

Clarke J.A., Mac Randal D. "An Intelligent Front-end for Building Performance Appraisal", *Proc. BEP '91*, Canterbury, April 1991.

Clarke J, "Assessing Building Performance by Simulation", *Building and Environment*, Vol 28, No. 4, pp419-427, 1993.

Clarke J, "A Performance Summary of Energy Conscious Building Technologies", accepted for publication in *Proc. EPIC '98*, Lyon, France, 19-21 November, 1998.

Crawley D., Personal Communication, Madison Wisconsin, 1995.

Hand J. "Thermal Optimisation Techniques for Traditional, Conventional and Passive Solar Building Designs Using EDT", *Proc. Renewable Energy Potential in Southern Africa*, Cape Town, pp16.1-16.17 September 1986.

Kaplan M. "Guidelines for Energy Simulation of Commercial Buildings", ACEEE Summer Study 1992. Vol 1. pp137-147, 1992.

Lawrie L.K., Crawley D.B, "What Next for Building Energy Simulation—a Glimpse of the Future", *Proc. Building Simulation '97*, IBPSA, Prague 8-10 September 1997, pp395-402, 1997.

Norton B. and Lo S.N.G., "Harmonising Environmental Simulation with the Building Design Process", *Proc. Building Environmental Performance '91* BEPAC, Canterbury, 10-11 April 1991.

Parand F, and Bloomfield D, "Quality Assurance in Environmental Performance Assessment", *Proc. Building Environmental Performance, BEP '91*, BEPAC, Canterbury, pp237-24, 6 April 1991.

Shneiderman B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Second Edition, Addison-Wesley Publishing Company, Wokingham, England, 1993.

Tufte E.R. *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut, 1983.

Tufte E.R. *Envisioning Information*, Graphics Press, Cheshire, Connecticut, 1990.

Tufte E.R. *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire, Connecticut, 1997.

Sonderegger R. "Building Performance Simulation in a Commercial Software Environment", *Proceedings BS '89*, IBPSA, Vancouver, BC, pp163-168, June 23-24 1989.

Chapter 5

INTRODUCING PROJECT MANAGEMENT TO SIMULATION

5 Introducing project management to simulation

> *"Should architects be significant users of mathematical models, or should these be the province of the consultant engineer?" [Howrie 1995]*

Chapter 3 identified constraints vis-a-vis the evolving nature of assessment questions during the design process, the complexity of the built environment, gaps in quality assurance tasks and the ability of simulation to work cooperatively with other design tools. Chapter 4 developed a specification which begins to address a number of these constraints. This chapter introduces a *project management* facility which implements this specification.

The **Project Manager** is an application which controls the process of simulation from the initial planning, through the phases of description, simulation, assessment and reporting. It allows a model to be composed in terms of objects identifiable to the user and with manipulative functions which support its evolution. The Project Manager has been designed to shift the point of nominal model complexity to something more in keeping with that of realistic design projects. It also allows those who do not fit the traditional definition of a simulationist to browse through and extract information from a model without having to know its underlying structure or the location of specific types of information.

The Project Manager has been implemented to deliver the objectives summarised below [Clarke 1992]:

1. The Project Manager should be the application that controls the data model and supports incremental problem definition (as required to support the design process) but does no more. It is therefore a "structure geared at letting other modules do the work".
2. It must control all problem files and the relationship between them although the code which reads and writes such data files will be common to all modules.
3. All other functions must be delegated: database management, data model analysis (e.g. consistency checking) and technical extension (e.g. view factor generation), simulation, results analysis, report generation, visualisations, etc.
4. The principal aim is to hide complexity by arranging for a single point of problem definition and evolution and a single simulator which can recognise partial problems (e.g. building only, flow only, etc.) and act accordingly. In this way we will be well able to support the many possible abstractions of reality that appear in the design process and so move the debate on to the key issue of which abstractions are valid and when they can be employed.

The Project Manager's aim is to ensure that an adequately trained and equipped team can make use of simulation to assess a design of arbitrary complexity and nature, without adversely affecting the design time-line or ad hoc nature of the design process. It is against these criteria that it should be judged.

5.1 Underlying structure

The design of the Project Manager is described in Sections 5.1.1 to 5.1.7 which encompasses: tight binding of the interface to the underlying data model, expression of the data model as objects in the user's domain, the containment of information related to all phases of a design project, an un-noticed interface, access to on-line support and a central desktop tool metaphor.

5.1.1 The central desktop tool metaphor

Simulation environments such as ESP-r have traditionally been implemented as a suite of applications. Within a multi-tasking computing environment such distributed functionality can be leveraged by knowledgeable users, but the autonomy of modules and the narrow focus of most user interactions have been found to be problematic.

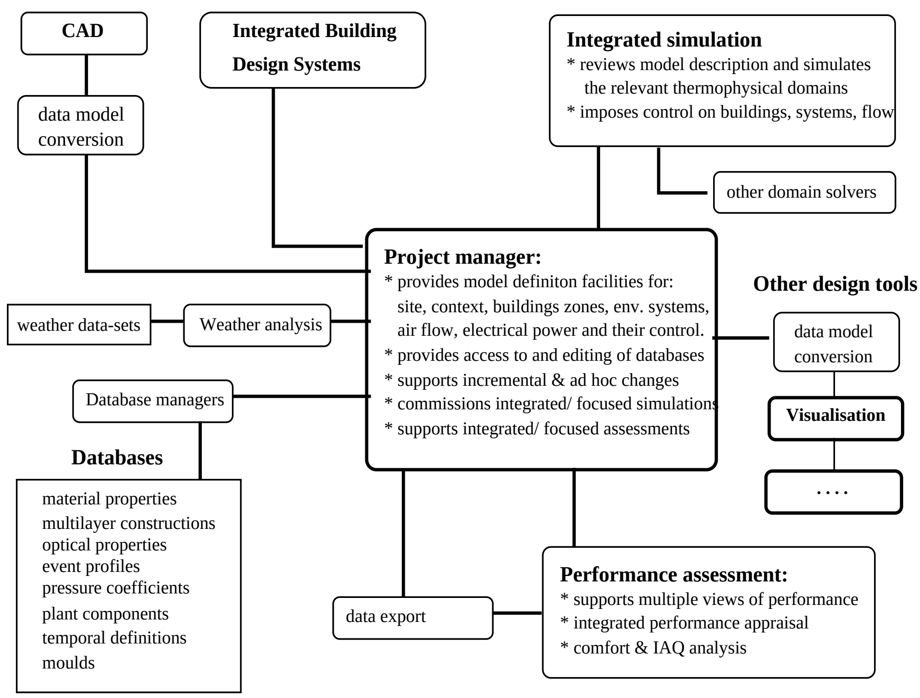


Figure 5.1 System diagram.

The approach taken in the current work has been to adopt the metaphor of a central desktop (the Project Manager) which acts as the entry point to all simulation tasks and the modules which support them, as shown in Figure 5.1. The desktop "owns" the simulation model and controls access to the various functions related to undertaking assessments. Control of the simulation process is based on the desktop's knowledge of the semantics and dependencies of the data model as well as of the functionality and dependencies of each module of the suite. Control is imposed by the way modules are invoked from the desktop, and by the nature of the information exchange between and within modules.

The implementation of the desktop is shown in Figure 5.2. The upper left shows the Project Manager which has invoked a simulation (lower left with several state-variables being monitored as the simulation progresses). A previous assessment is open on the lower right to allow the practitioner to judge changes in performance. In the upper right a visualisation is in progress. Each of these tasks was invoked from the Project Manager and each is active, but can be iconised to reduce screen clutter or brought into the foreground to, for example, check a component detail.

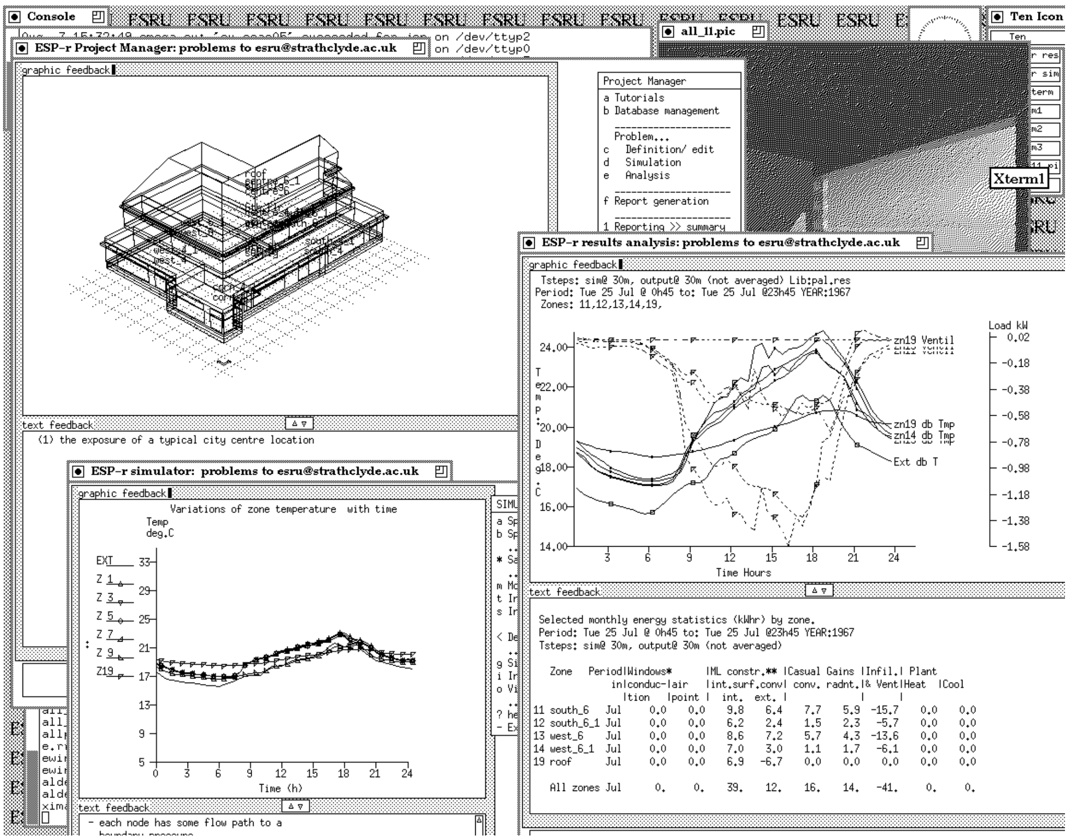


Figure 5.2 Work session with desktop and supporting modules.

To assist the invocation of support modules, a parameter-passing convention has been adopted wherein the name of the configuration, the type of user interaction (graphic or command driven), size, placement and optionally the focus (say a single zone) or intent (recalculate insolation patterns) are passed on invocation. Before exploring the functionality of the Project Manager the underlying design concepts are presented.

5.1.2 Tight binding of interface and data model

The design of the Project Manager is centred on the practitioner's relationship with, and access to, the data model. The need to tightly bind the interface and data model follows from observations of the efficacy of simulation when models were well matched to project demands (e.g. appropriately detailed, clearly structured and well documented). Recalling the data model entities described in Chapter 2, intervention is required to re-express the data model as a set of objects in the user's domain, with the detail, structure and documentation required to support design decisions and with facilities to resolve dependencies as the model evolves.

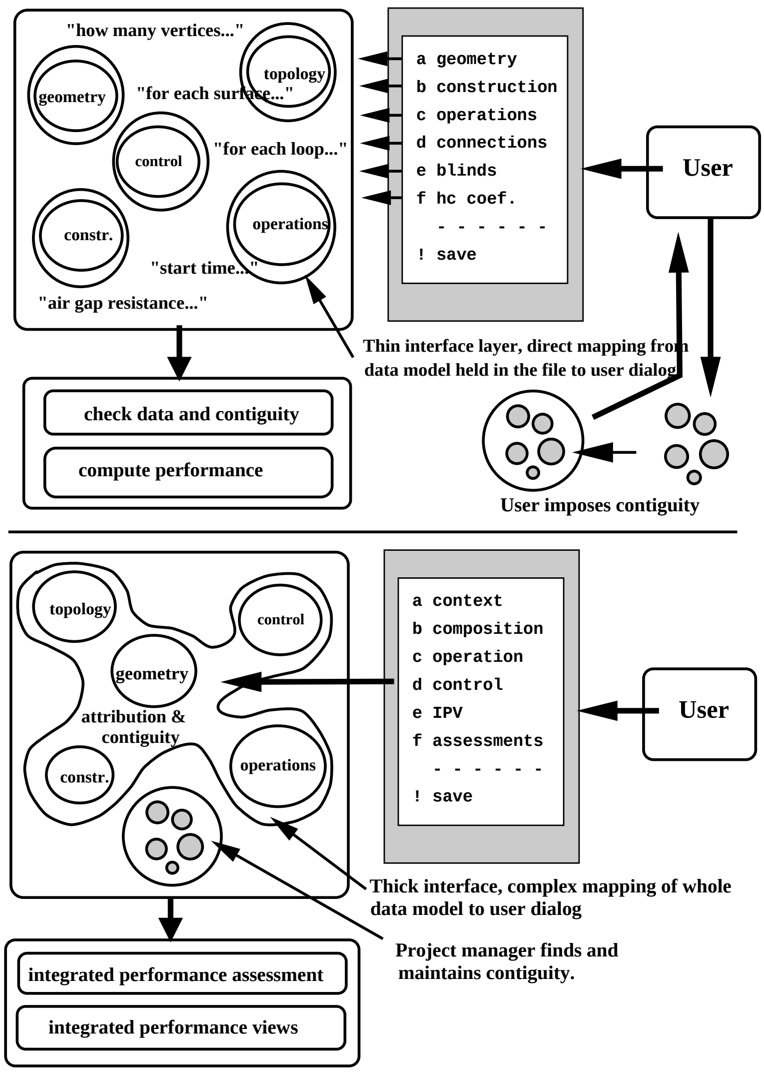


Figure 5.3 Autonomous modules with "thin" interfaces (above) and an alternative (below).

To clarify this, consider Figure 5.3. The upper half represents an autonomous approach (ESP-r circa 1990) where each task was focused on data file creation with a "thin" interface imposing few barriers between the user, data model and computational task. Contiguity was the responsibility of the user and became a software issue just prior to simulation. For those with a well defined mental map of the data model and simulation facilities, this allowed fine control of the simulation process and did not preclude undertaking assessments of considerable complexity.

However, in terms of the creation and evolution of models to support the design process, this regime was inappropriate on a number of counts. It acted as a barrier to the rapid and ad hoc evolution of models, imposed a steep learning curve and demanded attention to details and quality assurance issues which often distracted practitioners from the purpose of their work.

The lower portion of Figure 5.3 shows an interface layer supporting the expression of the data model as a set of objects in the user's domain. It reflects the current state of the model, the interconnections within the data model and the way understanding is enhanced by contextual information.

Implementing tight binding has required:

- the implications of a user action (e.g. a change in geometric resolution or database details) to be distributed to all relevant modules and that consistency be maintained within the data model;
- the context of an object to be included in displays, dialogues and reports;
- all attributes of objects and related databases to be accessible from the interface;
- the implications of the evolution of the model to be noted by the user and newly relevant assessment options brought "on-line".

For example, to increase the resolution of the model, the Project Manager might invoke a module to calculate the visibility matrix between surfaces. As these calculations are not dependent on contiguity or composition, this task may be undertaken as soon as the zone enclosure is defined, and updated as it evolves with little or no user intervention. Other tasks, such as the definition of flow networks and control, involve considerably more user interaction and the resolution of a number of dependencies within the data model and its file store.

Tight binding uses data links within a module and command links between modules to distribute the current model to other agents and to accept updates from them based on a set of agreed conventions. In terms of what the user sees on the screen, tight binding is concerned primarily with content rather than the specific format of display. Tight binding, as utilised in the current work, should be applicable to all simulation environments, even those with a single descriptive file or which are based on a monolithic application.

The coding implications of tight binding have required much of the code base of ESP-r to be redesigned. Clarity and consistency have been identified as critical requirements for simulation and this is also reflected in the design of source code. Changes have been made so that:

- selection/navigation tasks (e.g. copying or attributing a surface, changing focus from one zone to another) are based on selection from lists of user named entities;

- reporting detail (e.g. zone composition, schedules) can be set as silent, summary or verbose;
- feedback to the user (e.g. names of zones and surfaces, vertex indices) can be adjusted;
- mapping of logical tasks (e.g. invoking a simulation or a visualisation), tool functionality (e.g. call an intrinsic function, invoke an external agent), and consistency (e.g. addition of a surface invalidates the visibility matrix) are explicit and widely distributed;
- the file store is automatically annotated and supports unlimited user annotation;
- those functions which are shared by modules (e.g. reading from and writing to the file store) are held as common code;
- intrinsic functions (e.g. reading an integer from a file or user dialogue) include error trapping and reporting, range checking, default values and contextual help;
- low level functions (e.g. line drawing, user dialogue, list management, vector maths) are held in libraries and linked with modules for consistency;
- functions and subroutines conform to a regular coding style in terms of how they pass parameters, trap errors and treat arrays and are commented;
- the data model is strongly typed and structured, uniformly expressed in all modules and formally defined.

This has been implemented by partitioning the code into application-specific code, a body of common code and a library of low-level functions. Overall the ESP-r system includes 245,000 lines of application source code, common code and library code and comprises 1750 subroutines. The Project Manager code (excluding common and library code) is some 40,000 lines in 182 subroutines.

The need for a common code base stems from the number of functions and facilities which act on the data model within the ESP-r suite, and from the demand for rapid prototyping. For example, reading the geometry and attribution of a zone from the file store is a distributed task. Modules importing a single facility are consistent and inherit any evolution in that facility. There are some 173 common Fortran functions and subroutines held in 22,300 lines of common code. The use of common code restrains the overall growth of coding. For example, the application code for the surface contiguity checking module is 2381 lines. This expands to 13,000 lines (excluding library code) at the time of compilation.

Functions which require wide distribution, but do not act directly on the data model are held in a library which is linked with each module. The library comprises some 6,400 lines of Fortran in 244 functions and subroutines. There are also 6,000 lines of C code in 81 functions. Many of the C functions act as a bridge between high level graphic instructions and machine dependent graphic code.

5.1.3 Object representation and manipulation

One way to envisage a simulation model is as a set of building blocks (e.g. surfaces, zones, controls, components) which have been assembled to represent in abstract the essential characteristics of a design. Simulation presumes no particular scale or level of geometric detail (e.g. a model of a thermostat housing and a model of a shopping mall can be composed from the same entity types in the data model). The approach taken has been to apply object treatment both at the level of individual components and to the collection of entities which form a project. Consider the building definition facility in Figure 5.4.

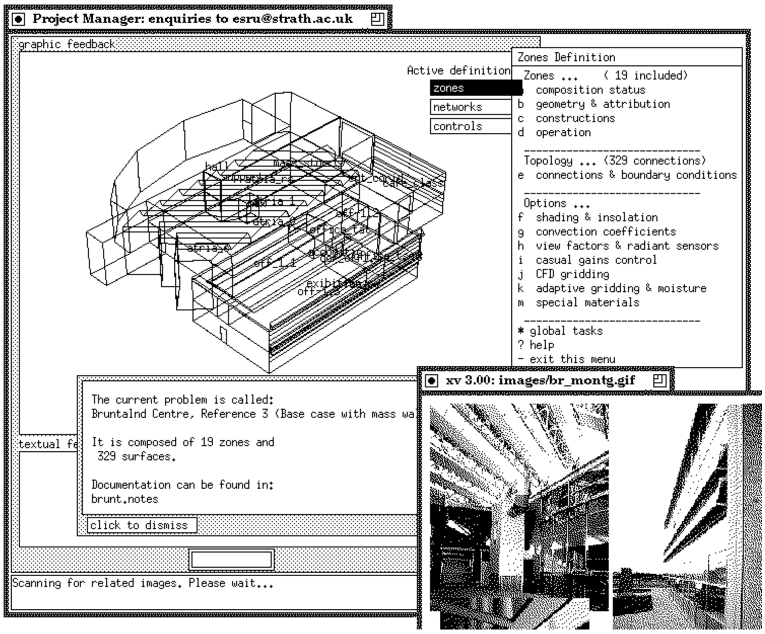


Figure 5.4 Building definition facility.

After selecting the zones button (highlighted) "Active definitions" the user is presented with a synopsis and a relevant image (based on a directed traverse of the data model). Selecting "geometry and attributes" would shift the focus to one thermal zone and its attributes. Similar facilities exist for "operations" (schedules of occupancy, small power loads and infiltration/ ventilation), or extensions to the building definition (shading patterns, imposed convective heat transfer coefficients, etc.).

Object treatment includes both the presentation of all attributes of an entity and its context. Consider the zone definition facility shown in Figure 5.5. Not only are zone and surface names included, but the display on the right includes supporting information defining volume, floor area and position in space. The context of the information in the zone composition report (bottom left of Figure 5.5) is clarified by being one of many views of the data model. The user may access the vertices and the edges which make up the enclosing polygons, as well as the composition and topology attributes which give the surfaces thermophysical meaning.

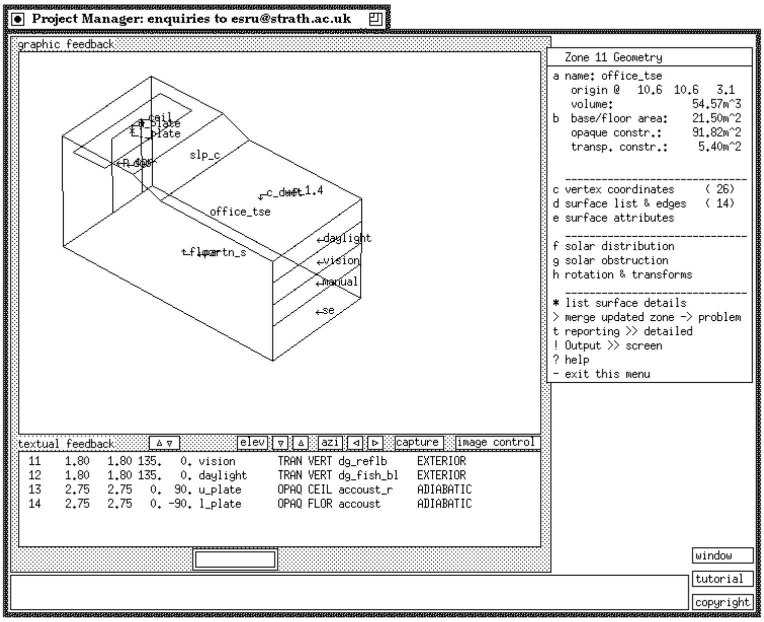


Figure 5.5 Display and report of the salient features of a zone.

In combination, the text, graphics and reports form a compact archival definition. This pattern ensures clarity in selection tasks (see left of Figure 5.6) and in the manipulation of surfaces (see right of Figure 5.6). Here all the attributes of the surface (and of its boundary conditions) are displayed and can be modified (as can each of the databases which contributes to the surface definition).

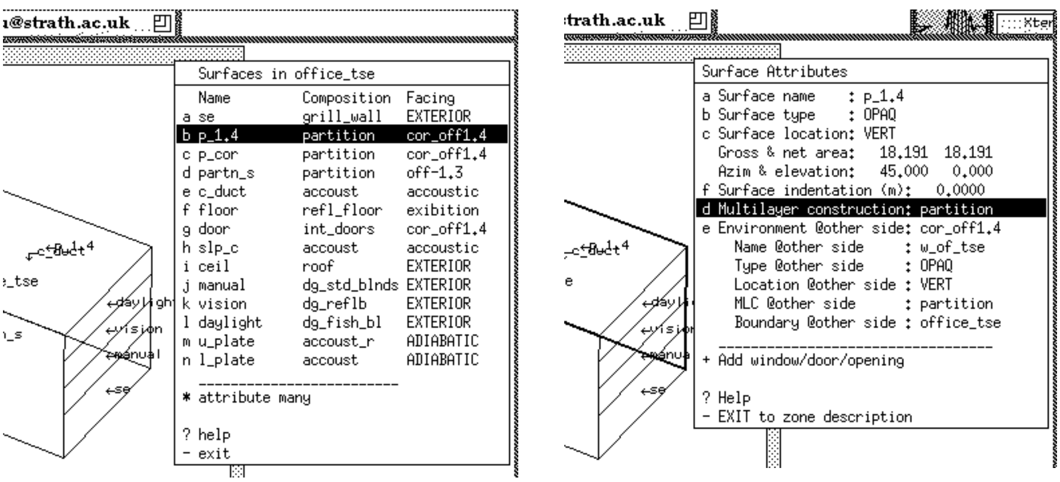


Figure 5.6 Object treatment of surface list selection (left)
and surface attribution (right)

Figures 5.5 and 5.6 also show facilities for adding, deleting, copying and transforming entities, with dependencies resolved. Such facilities should not be confused with automation of the descriptive process. The intent is to clarify and make explicit the relationships within the model. Because of this, the implementation of these manipulation facilities has been tempered by consideration of:

- the extent to which dependencies can be resolved without user consent;
- the frequency with which user interaction is required (e.g. to confirm deletions) to limit error propagation without becoming tedious;
- the risk of isolating the user from important aspects of a model or the underlying thermophysical processes (where such tasks are carried out silently, users seem detached from their models and unaware of discontinuities).

Although the data model is static, there are opportunities for the user to introduct new concepts and associations. An example of this is a facility to associate a list of surfaces with a named concept, say "external glazing" and then manipulate those parts of the model associated with the concept. The Glazing Design Support Tool (GDST) [LESO-PB|EPFL 1998] developed within the IMAGE project uses this to update simulation models based on a designers request to test alternative fenestrations.

The treatment of entities as objects in the user’s domain is not limited to the descriptive process. Clarke [1998] uses the phrase "behaviour follows description" to characterise how the *ad hoc* and incremental nature of the descriptive process gives rise to incremental access to assessment functionality. It is no longer the case that simulation requires a substantially complete model to deliver information to the design process. The Project Manager recognises the information requirements of specific assessment tasks and as the model crosses such thresholds new options become available. Further aspects of object expression are explored in the discussion of interface design (Section 5.1.5).

5.1.4 Containment of project data

The case studies provided evidence that the conventional focus of the simulation data model on the composition of buildings and systems is a constraint. What is required is the containment of project information. Although tight binding effectively hides the detail and structure of the underlying data model, is is also critical to re-express the data model in ways which support project demands.

As implemented, project information includes links to databases, images, reports, project documents and logs. To support access to these varied and distributed information sources, the data model also records the agents which can deal with each type of information as well as the message passing conventions and any filtering or transforms which may be required. For example, the inclusion of project images can greatly enhance the clarity of a model. One successful technique (illustrated in Figure 5.7) is to use montages providing an overview of the project, the model composition and, optionally, a synopsis of the results of the project.

The data model has also expanded to hold documentation related to specific contexts (e.g. notes, assumptions, references which were previously relegated to scraps of paper). The ability to retain phrases such as "operating theatre 7ac/h assuming standard clinical team, $40W/m^2$ equipment with 2ac/h and $10W/m^2$ standby 23h00 till 6h00" to identify one of six occupancy regimes to be tested, is

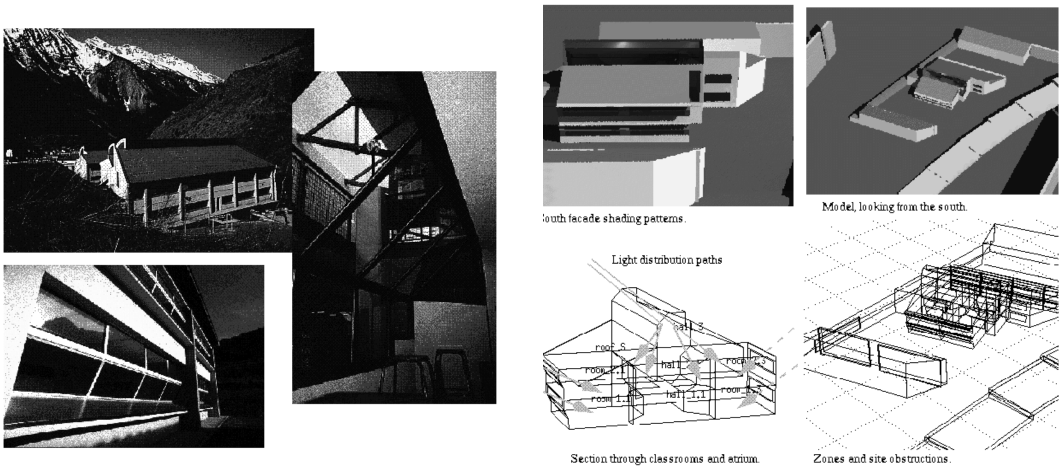


Figure 5.7 Project montages.

often critical to project deliverables.

The file store (see Sections 2.1 and 2.2) is both typed (i.e. zone schedules are held separately from zone composition) and distributed (i.e. systems control and zone definitions are held in separate folders). A partitioned data store has several advantages, but can be chaotic if its structure is not clearly delineated. The binding agent is the *project configuration* which is the core encapsulation of the current state of the project. Project configurations are created, maintained and evolved by the Project Manager and form the basic unit of information exchange. A portion of a project configuration that deals with the project context is shown in the upper portion of Figure 5.8.

The extended nature of the data model is expressed in the system configuration file. For example, to assist information management, the names and locations of folders which contain zone, network, control and visualisation related information have been included. Macro scale issues (e.g. the selection of fuel types) are supported by an environmental emissions section. More importantly, the metrics of the project are encapsulated in an Integrated Performance View (IPV). Such issues are explored in detail in Section 5.3.

The balance of the project configuration is given over to a list of files for each zone, each tagged to indicate the nature of its contents. While a basic assessment requires only geometry, composition and operational information, detailed analysis may require imposed convective heat transfer coefficients, surface view factors, CFD domain definition and 3D conduction gridding schemes. The "Building" portion of a configuration file is shown in the lower portion of Figure 5.8. Individual entries and formatting conventions are described in Appendix D.

In terms of incremental instantiation of a model, this regime offers several advantages. Firstly, many assessments require project configurations which are minor variants of a base case, to test different design or control options, for example. The use of file references supports step-wise modifications to a single project configuration, or the option to create a set of project configurations. Secondly, the distributed nature of the file store allows parametric variations of a single issue (e.g. different occupancy regimes in a zone) to be expressed explicitly while holding all other aspects of the model constant.

```
* CONFIGURATION3.0
# ESRU system configuration defined by file
# brunt_pvbcc.cfg
*date Tue Jun  9 20:51:09 1998  # latest file modification
*root brunt_pvbcc
*zonpth ../zone_pvbc              # path to zones
*netpth ../nets                   # path to networks
*ctlpth ../ctl                    # path to controls
*imgpth ../images                 # path to project images
*indx    1 # Building only
 55.000   -3.000   # Latitude & Longitude (diff from meridian)
      6   0.200   # Site exposure & ground reflectivity
* DATABASES
*prm  ../database/brund_prm.db1       # materials
*mlc  ../database/brundtland.mlc      # constructions
  . . . [cut section]
*clm  ../database/danish.try
*ctl  ../ctl/pvbc.ctl
*vew    -40.0  -100.0   100.0    -0.9    24.3     4.0 40.0
*year  1999 # assessment year
*img GIF   ****  ../images/br_montg.gif
*img GIF   FZON  ../images/br_model.gif
# prim energy conv (heat,cool,lights,fan,sml pwr,hot water)
*pecnv  1.050 1.730 1.730 1.730 1.730 1.050
*htemis   190.000    0.300    0.200 # heating emissions CO2,NOX,SOX
  . . . [cut section]
*hwemis   190.000    0.300    0.200 # dhw emissions CO2,NOX,SOX
*ipv  ipv/pvbc.ipv
* PROJ LOG
brunt.notes
  . . . [cut section]

* Building
Brundtland Centre, Reference 3 (Base case with mass wall thermal store)
     19  # no of zones
*zon   1   # reference for ent_corid
*opr ../opr/entrance.opr  # schedules
*geo ../zone_pvbc/u_corid.geo  # geometry
*con ../zone_pvbc/ent_corid.con  # construction
*tmc ../zone_pvbc/u_corid.tmc  # transparent constr
*zend
*zon   2   # reference for cafe_class
*opr ../opr/cafe_class.opr  # schedules
*geo ../zone_pvbc/cafe_class.geo  # geometry
*con ../zone_pvbc/cafe_class.con  # construction
*tmc ../zone_pvbc/cafe_class.tmc  # transparent constr
*isi ../zone_pvbc/cafe_class.shd  # shading db
  . . . [cut section]
*zend
*cnn  brunt_pvbcc.cnn  # connections
```

Figure 5.8 Configuration header and building composition section.

The extension of the data model and the tight binding of the interface to the data model enhance the understanding of models. Managers of simulation teams are thus in a better position to track the progress of their staff, prepare reports and present material for discussions with clients. The extension of the data model also takes into account the temporal nature of projects in terms of new information becoming available and new projects starting. The latter implies that it must be possible to set up a project configuration before any building details are known. The approach taken to make this so has been to introduce a "registration level" to the descriptive process (see Figure 5.9) so that the desktop becomes an active part of the design process, even before the form and composition of the zones are apparent. The registration of a project includes setting up the project folders so that future documents, databases, images and zone data are held consistently.

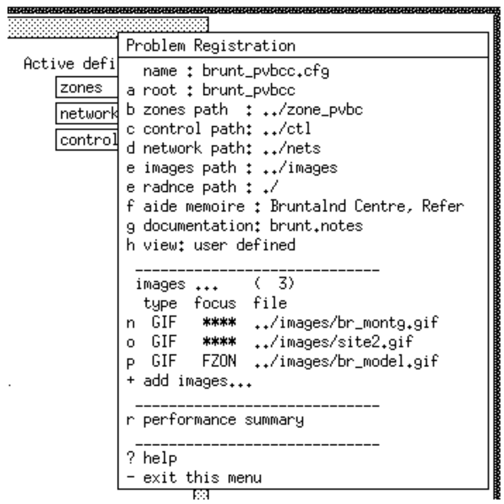


Figure 5.9 Project registration facility.

5.1.5 Data model transforms and exchange

The exchange of simulation models, described in Section 2.3 and 4.2, is an issue in the design of the Project Manager. Although superset to subset model exchange, such as the hidden-line viewing application (see Figure 2.5) are unremarkable, the bi-directional exchange of geometric information with CAD is more complex and is subject to data loss. A typical exchange between ESP-r and the CAD application Xzip is shown in Figure 5.10. Although the geometric translation is superficially straightforward because of the similarities in data models, Xzip files do not include topological or compositional attributes so there is data loss implicit in any exchange.

Exporting geometry to a subset data model exchange medium such as DXF is also straightforward. The approach taken is to overload DXF's layering convention to place opaque and transparent surfaces and obstructions in each zone into separate layers (e.g. office_glz, office_opq, office_blk, corid_glz, etc.). Surfaces and obstructions are converted into 3D faces and polygons, but names, attributes and topology are lost (see Table 4.1 for data model differences). Unfortunately, the reverse transform is viable only if the same conventions were used in the creation of the CAD model. Even with this, the user is still required to attribute and define the topology of all imported surfaces.

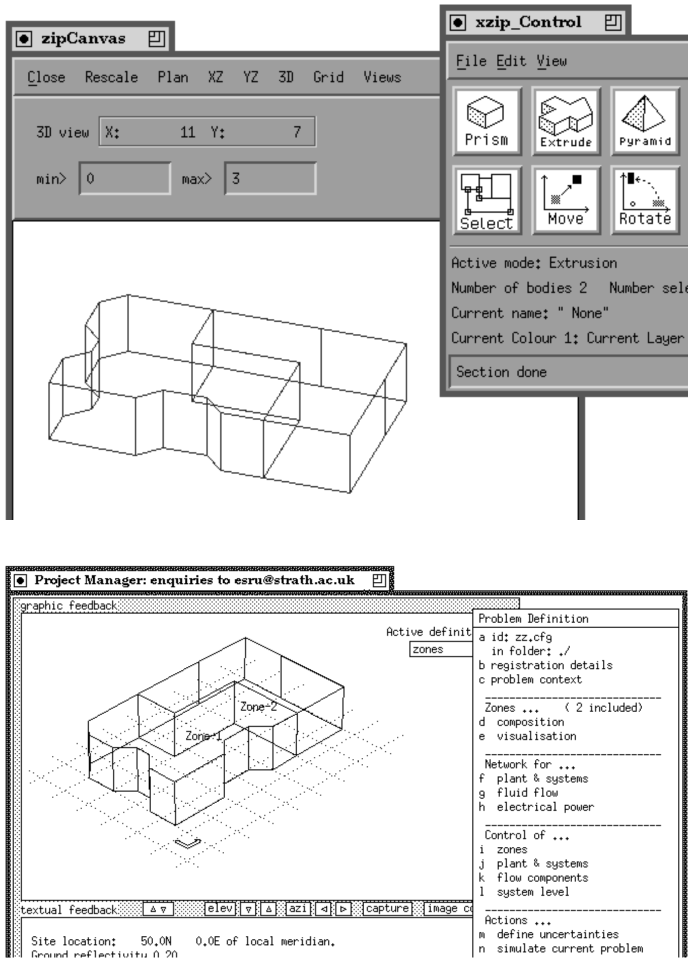


Figure 5.10 Transform between Xzip and ESP-r.

Filtering between simulation tools can be done only on the basis that the semantics of the data models are not compromised during the conversion process. In the case of ESP-r to TSBI3 [Johnsen and Grau 1994], this has been made possible by joint work by the author and the TSBI3 development team. Models can be derived from data structures within the code or from the file store. Given that ESP-r uses a distributed file system and TSBI3 a monolithic file store, the most straightforward translation proved to be from data model (in memory) to TSBI3 file.

One of the consequences of the lack of CAD attribution has been the extension of CAD facilities within the Project Manager. These have taken the form of transforms, rotations, mirroring and replication of geometric entities and the ability to copy, move or remove entities as required. Topological checks are also undertaken more frequently and attribution can be done by exception.

5.1.6 Implementing an un-noticed interface

> *A good interface is the foundation of a program rather than a finishing touch [Sonderegger 1989].*

The perception that a polished interface will empower the user is particularly prevalent in PC tools where the relative ease by which complex displays can be implemented has resulted in their proliferation [Shneiderman 1993]. However, simulation may not be well served by slavish attempts to keep abreast of trends in human-computer interfaces. Firstly, commercial interfaces are particularly demanding of development resources [Beranek and Lawrie 1989, Crawley 1997]. Secondly, excessive functionality leads to clutter, complexity and a host of training and support difficulties [Bailey, 1989].

The "rules of engagement" for the Project Manager interface, as stated in Section 3.6.5, are founded on the issues of clarity and consistency. A number of interaction rules have been defined and are listed in Table 5.1. Some interaction rules appear simplistic: information should be entered only once, it should be difficult to select an inappropriate option, it should be possible to find out more about the current task including a reasonable default value, and common or repetitious tasks should require minimal interactions. The challenge is to consistently apply such rules.

Table 5.1 Interaction rules.

Convention	Implementation	Result
Express and manipulate entities as objects in the user's domain.	Design interface and underlying structures to allow consistent entity manipulations.	Users pay less attention to bookkeeping and have fewer barriers to understanding models.
Simulation tasks accessible via logical commands and lists of entities named by the user.	A central desktop translating user requests into the command syntax of the modules. Generic list selections.	Allows users to delay detailed definitions. Opens simulation to casual access. Clearer presentation of the model.
Dialogue consistent and well supported with a single point of interaction and a single focus.	Common library calls for all dialogue, context help, defaults and range checking provided throughout.	Once patterns established, easy to expand skills. Reduces barriers to intermittent use.
Feedback designed for clarity. Redundant feedback via combinations of names, attributes and images. Dynamic displays and selection mechanisms reflect the current state of the model.	Simultaneous display of text, model attributes and images. List selection based on user provided names.	Selection and manipulation with fewer errors. Less effort on the part of the user to understand the model.
Access to exemplars demonstrating appropriate syntax as well as how simulation can support the design process.	Facilities to select and load remotely held models. Creation or upgrading of models to form annotated exemplars.	Allows users to proceed at their own pace to build a mental picture of where various facilities lie within the system.

Convention	Implementation	Result
Ad hoc access to tutorials about the interface, facilities offered in various modules, the data model and guidance on the composition of models.	Dialogue supported by context-sensitive help. Enabled both on-line and hypertext based tutorials.	Mixed results for context help and on-line tutorial. Hypertext/ WWW tutorial, proved useful in workshops and in support of distance learning.

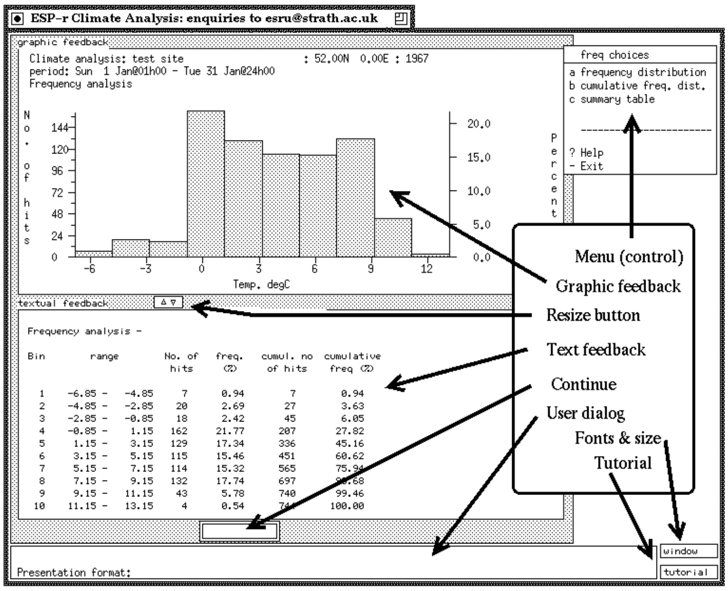


Figure 5.11 Interface elements.

All modules use a similar layout (see Figure 5.11) with specific areas for control (menus), text and graphic feedback and, at the bottom, user dialogue. Each control menu in each module includes a "? help" entry which provides a synopsis and (sometimes) supporting information as shown in Figure 5.12. Table 5.2 gives a synopsis of the common interface elements.

Many interfaces make use of pro formas, especially where there is a cluster of related entities. This imposes a considerable code overhead and so similar functionality has been achieved by overloading the control menu as shown in Figure 5.13. Editing of entities at their point of display, rather than in a dialogue box, would require less mouse and eye movement, but only by complicating the provision for a default and context-sensitive help. As there are literally hundreds of dialogues, the overhead and the possibility of visual clutter argued against such an approach.

The menu conventions adopted for consistency are of first column key selection (which allows modules to also be run in text mode) followed by a label and then the item. If no key selection is included the entry is for feedback only. A ":" after the label shows the item or items to be edited, a ">>" indicates a toggle. Horizontal bars or dots are used to separate groups of entities. The ordering of tasks is predominantly top to bottom with the last two entries for help and return selections.

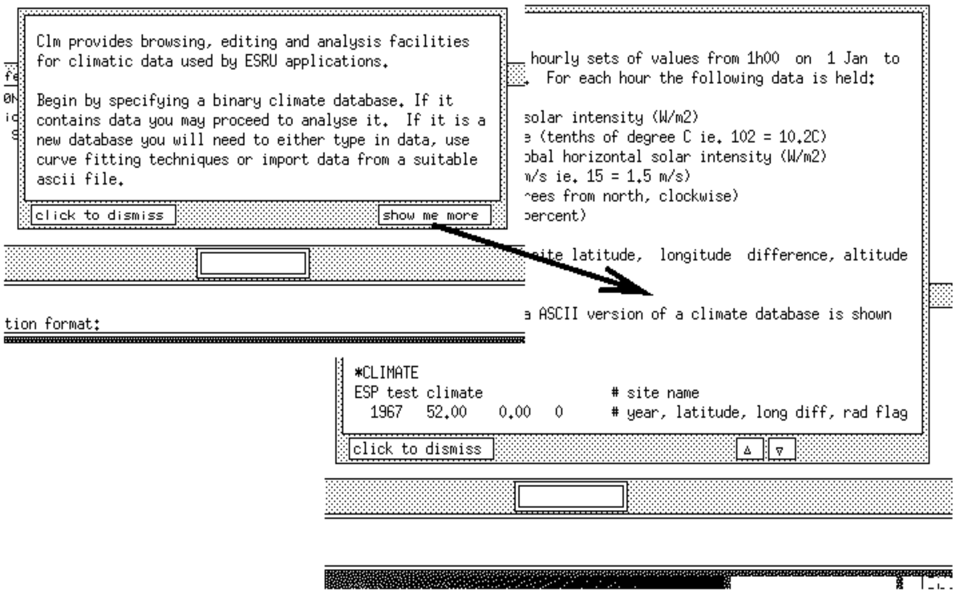


Figure 5.12 Help/ synopsis of command menu and module features.

Table 5.2 Interface elements.

Element	Description	Constraints/ Rules/ Syntax
Text feedback	Tabular information such as lists of attributes or reports.	One way scrolling.
Dialogue	Standard dialogues (see Figure 5.14) with defaults, context help and prompts supported.	All dialogue in same location, right justified.
Pop-up text display	Displays context help for user dialogue or advisories.	1-60 lines, scrolling, dismissed with click.
Tutorial button	Invokes hypertext based browser.	Included in all modules.
Environment button	Allows font selection and window refresh/resize.	Included in all modules.
Continue button	Linked with text feedback (to invoke scrolling) and dialogue to signal continue after warning messages.	Click to continue.
Resize button	Drag up:down to change relative size of graphic and text feedback.	Drag and release.

Selecting "b" in Figure 5.13 would invoke a period dialogue, after which the user's attention would again be focused on the climate control menu.

Because ESP-r is implemented in a UNIX® environment, remote invocations with a graphic interface are possible. Provision has also been made for modules to be invoked in text-only mode (Figure 5.15). This makes remote access possible even with limited communications bandwidth. It is also helpful for managers who wish to check the current status of a project from a remote site.

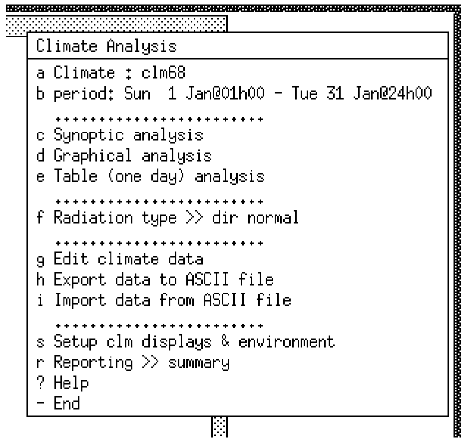


Figure 5.13 Overloaded control menu.

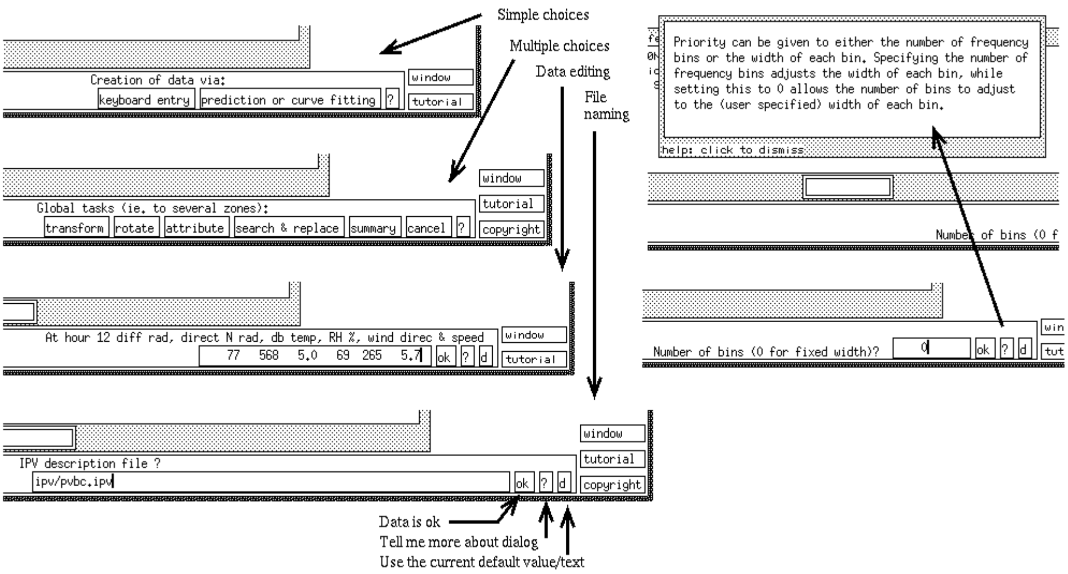


Figure 5.14 Standard dialog forms and context-sensitive help.

The conventions described result in a desktop which is not static, i.e. the interface changes to provide feedback on the current state of the model. The user is able to select an item, modify it, see the effects of the change, access context-sensitive help to review aspects of the item, and then proceed as required. With a few keystrokes the composition of any surface, control or network component is available. There is a consistent way of providing information, consistent dialogue style, editing and control. These are interface essentials which could be supported by any number of interface forms other than the pragmatic form of the current implementation. What is gained by such pragmatism? As Clarke has remarked on numerous occasions "never mind the interface, consider the functionality". The case study in Chapter 6 provides evidence of the efficacy of the approach taken.

```
Browse/ create problem using:
  a) Project Manager, b) CAD, c) continue ? a

 Zones Definition:  Zones ...   ( 19 included)
                   a  composition status
                   b  geometry & attribution
                   c  constructions
                   d  operation
                    _____________________________
                    Topology ... (329 connections)
                   e  connections & boundary conditions
                    _____________________________
                    Options ...
                   f  shading & insolation
                   g  convection coefficients
                   h  view factors & radiant sensors
                   i  casual gains control
                   j  CFD gridding
                   k  adaptive gridding & moisture
                   m  special materials
                    _____________________________
                   * global tasks
                   ? help
                   - exit this menu
 Zones Definition:?>
```

Figure 5.15 Text interface.

5.1.7 On-line support, exemplars and audit trails

The case studies identified three aspects of training which require software support. The first is access to on-line tutorials which bridge the gap between context-sensitive help and traditional users' manuals. The second is access to well documented and structured exemplar problems for use as instructional aids. The third is to record user actions (e.g. decisions taken, in-built and external facilities invoked, context-sensitive help accessed) so that an audit trail is established.

On-line tutorials

To support the first demand, and to support the testing of alternative presentation techniques and tutorial contents, a "tutorial" module was developed for use within the Project Manager and other modules. A typical screen is shown in Figure 5.16.

The mechanism chosen to display the tutorial material was a straightforward one. Upon selection of a topic from the menu hierarchy, a unique key is passed to a search/display engine. The key is located within a set of tutorial files and the contents displayed. The simplicity of the search engine and text display allowed for testing different approaches to on-line support and the bulk of the development resource to be given over to the content of the tutorial. The latter was of particular concern as much of the literature on instructional design cautions against direct transcription of manuals, primarily because of low display resolution and the fatigue encountered in scanning through blocks of text. This demands smaller information "chunks" as well as attention to the tutorial menu/navigation hierarchy. In the event, it was possible to remove verbose text from materials included in the tutorial. It was also the case that the tutorial allowed some topics to be expanded. For example, methodology and quality assurance practices, which were poorly represented in written documentation, were linked to example

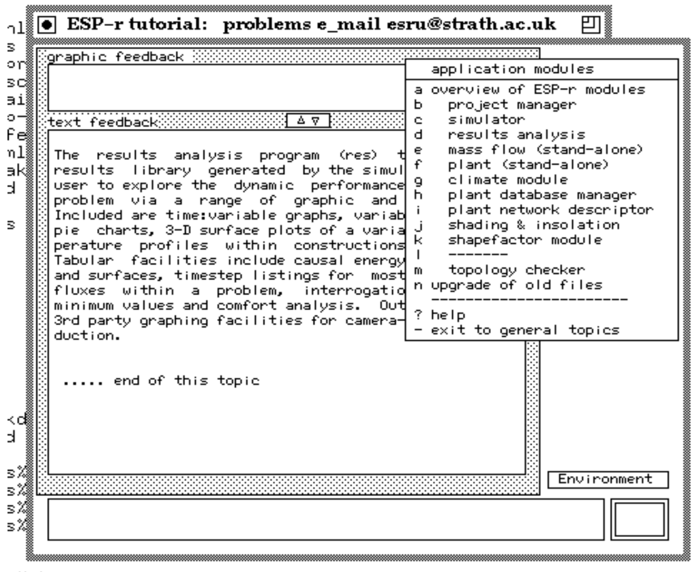


Figure 5.16 Tutorial facility.

simulation models within the tutorial.

Hypertext tutorials

During the development of the on-line tutorial, hypertext browsers such as Netscape <http://www.netscape.com> became available on the World Wide Web (WWW). As part of the current research, the more successful portions of the tutorial were translated into HTML format to see if this was a viable medium for supporting simulation skills acquisition. HTML supports topic hierarchies and enhanced topological conventions (text of different sizes and emphasis, paragraph re-formatting, higher information densities, etc.) as well as images and links to related subjects.

This Web based tutorial was one of the first attempted in the simulation community, and coincided with the considerable debate in academic circles [Parrington 1994, Sangster 1995, Zhao 1996] regarding attempts to fit the language of pedagogues and etymologies [Pickering 1995] into a system initially designed to support the cooperative work of scientists at CERN. Hypertext based facilities have since matured to the point of ubiquity. A montage of hypertext pages are shown in Figure 5.17 and further detail on coverage of specific topics and the data model is shown in Figure 5.18.

The initial audience of undergraduate and postgraduate students found hypertext to be an improvement over the tutorial. In particular, the inclusion of images help clarify relationships within the data model and links to exemplars enhanced the presentation of methodology and abstraction issues. The density of screen information, coupled with the speed and flexibility of navigation made it possible for students to access more topics within a given time frame. Based on this initial feedback, the Web page was advertised to the ESP-r community in March of 1995 and the first steps were made to co-ordinate the WWW pages with the existing coursework.

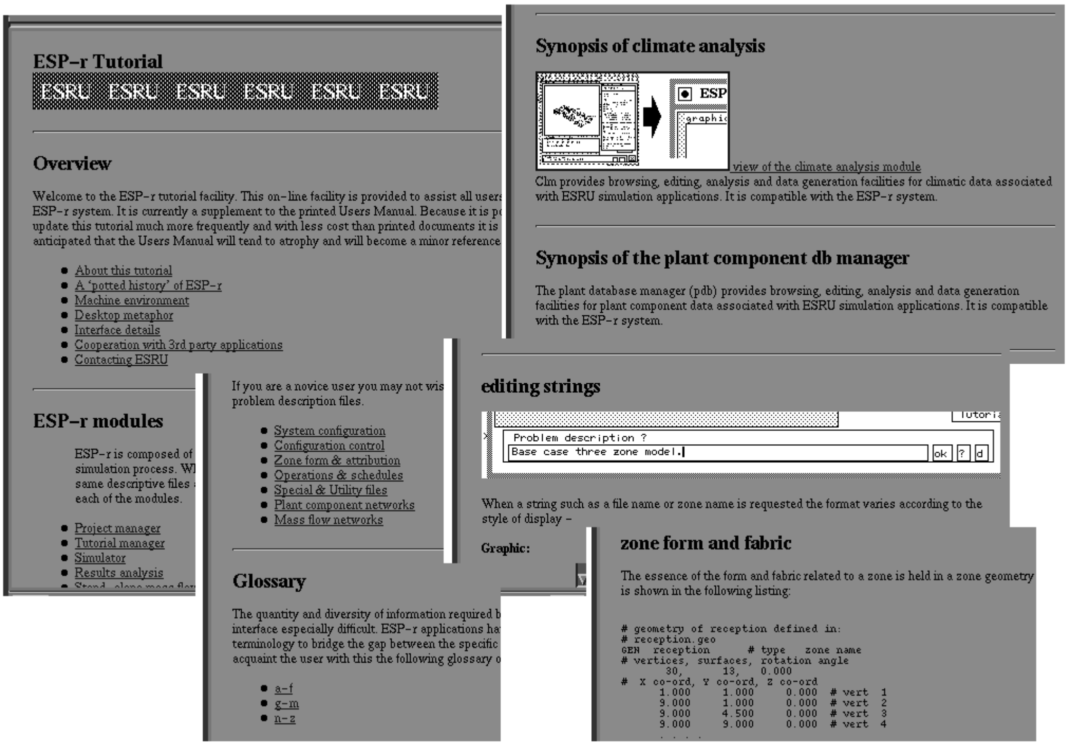


Figure 5.17 Montage of 1994 WWW tutorial.

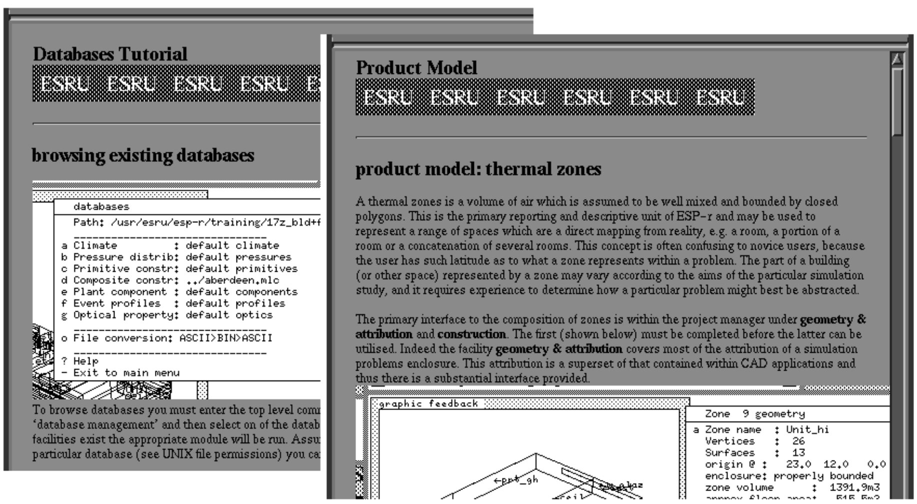


Figure 5.18 Hypertext Database support and data model details.

Exemplars

The limited persistence of models and the general inability of simulation teams to reuse, let alone replicate studies which were more than a few months old is a barrier to simulation use. That models could be transparent (in the sense that the project intent, methods and abstraction were clear) and transportable (in the sense that there was a reasonable expectation that the model would function on another machine and with another simulation team) were central issues driving the evolution and testing of the Project Manager. It was also a goal that transparency and transportability should be attainable by models at both extremes of geometric and operational complexity.

The approach taken to achieve this recognises that evolving concise approaches to the support of design questions, and ensuring that others may carry forward such work, involves a mix of clarity, consistency and an extended data model on the part of the tool, and method and art on the part of the user. Providing facilities for logical naming schemes, supporting dispersed documentation (e.g. for specific and general facets of a model) and ensuring that reporting facilities deliver information in forms which make sense in a design setting are examples of coding interventions.

Exemplars allow opinions to be formed on "what constitutes best practice" from scores of well documented realistic examples of simulation support for the design process rather than extrapolating from contrived examples in a users' guide. Exemplars allow instructors to compose exercises to "...explore the naturally ventilated office exemplar and gather information on the following design parameters...". In the context of advanced training, exemplars allow users to bypass the tedium of the descriptive process so that attention could be given to how, for example a PID controller can be tuned. As each user is working with the same exercise and accessing the same documentation it is easy to identify those having problems. That most participants were able to understand a model they had no previous experience of within a half hour demonstrates the efficacy of this approach. In the context of a research or consulting project, exemplars might include documentation on measured performance or post-occupancy evaluations.

As the number of exemplars increased it became clear that a hierarchical selection mechanism (Figure 5.19) was required. This was supported by information held in an "exemplars database" which could be configured as required. The conventions and mechanisms employed in exemplars are discussed in Appendix D.

Audit trail

The limitations of direct observation of users suggest that some other agent should observe the use of simulation. The approach taken is to record an audit trail of the user's actions and decision points in an electronic journal. Each journal entry is time stamped and given a key word to aid in subsequent scanning for particular actions. This is accomplished by inserting statements in the form: `call tstamp('>','PRJ: edited zone name')` within the source code at points where decisions are taken, interactions invoked or the model altered. Figure 5.20 shows a portion of a journal which indicates that the opening of an existing problem, generating a high level report, updating registration details, focusing on one zone, checking details and then altering the attribution of a surface and updating the model. This facility has implications for project management, quality control, skills acquisition as well as providing useful information to the author on how facilities are used in practice.

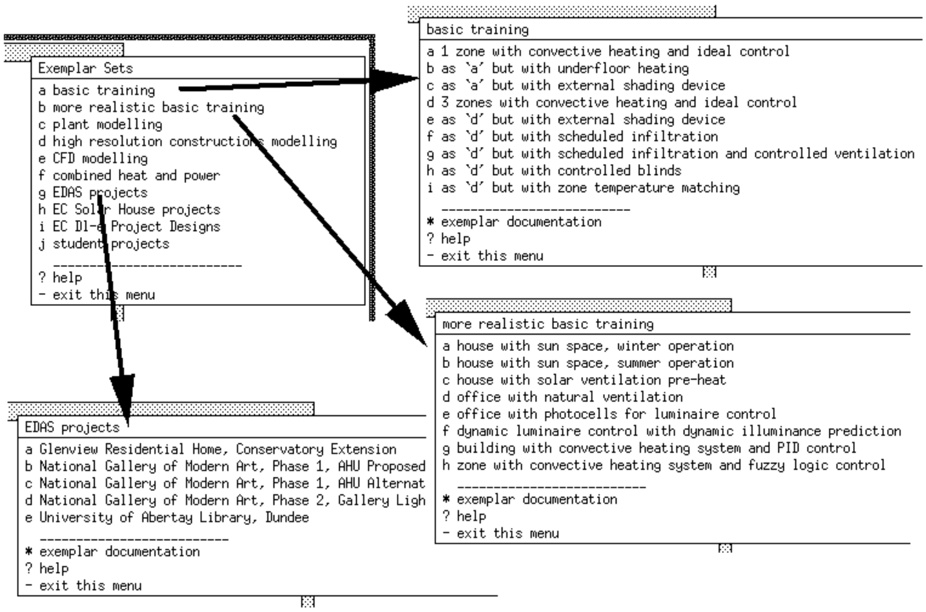


Figure 5.19 Sets of exemplars.

```
Journal for:jwh
Date: Tue Jun  9 19:33:28 1998
Project reference:
PRJ: loading supplied problem @ Tue Jun  9 19:33:28 1998
brunt_pvbcc.cfg @ Tue Jun  9 19:33:28 1998
PRJ: global summary @ Tue Jun  9 19:37:09 1998
PRJ: update registration @ Tue Jun  9 20:37:04 1998
PRJ: problem context @ Tue Jun  9 20:38:42 1998
PRJ: focus on office_tse @ Tue Jun  9 20:46:50 1998
PRJ: list zone surface summary @ Tue Jun  9 20:49:26 1998
PRJ: enter zone vertices @ Tue Jun  9 20:51:04 1998
PRJ: enter single surface attribution @ Tue Jun  9 20:52:59 1998
PRJ: update zone geometry @ Tue Jun  9 20:53:41 1998
PRJ: zone summary @ Wed Jun 10 08:01:56 1998
 . . .
```

Figure 5.20 Extract from audit trail.

5.2 Project Manager functionality

The previous sections explored the use of a central desktop tool metaphor, tight binding of the interface to the data model, object expression of the data model and the containment of project information. In this section the functionality of the Project Manager is reviewed. This discussion is presented roughly in the order of tasks required to compose a model. It begins with database management and proceeds through the definition of context, building, zone, networks and control.

5.2.1 Database access

One of the first tasks in any simulation project, and a core issue in the management of projects, is to establish relevant databases, either via links to corporate resources or by the creation of project specific data. To reflect this, database access and maintenance facilities are among the initial options within the Project Manager. As shown in Figure 5.21, each of the databases can be accessed from a central list for browsing or maintenance. In some cases such functionality is provided by support modules. The climate analysis tool, for example, provides analysis and display as well as database creation and editing facilities. In the lower portion of Figure 5.21, selection of an optical entry has provided a graph of the relationships between the various angular properties as well as editing fields and import facilities.

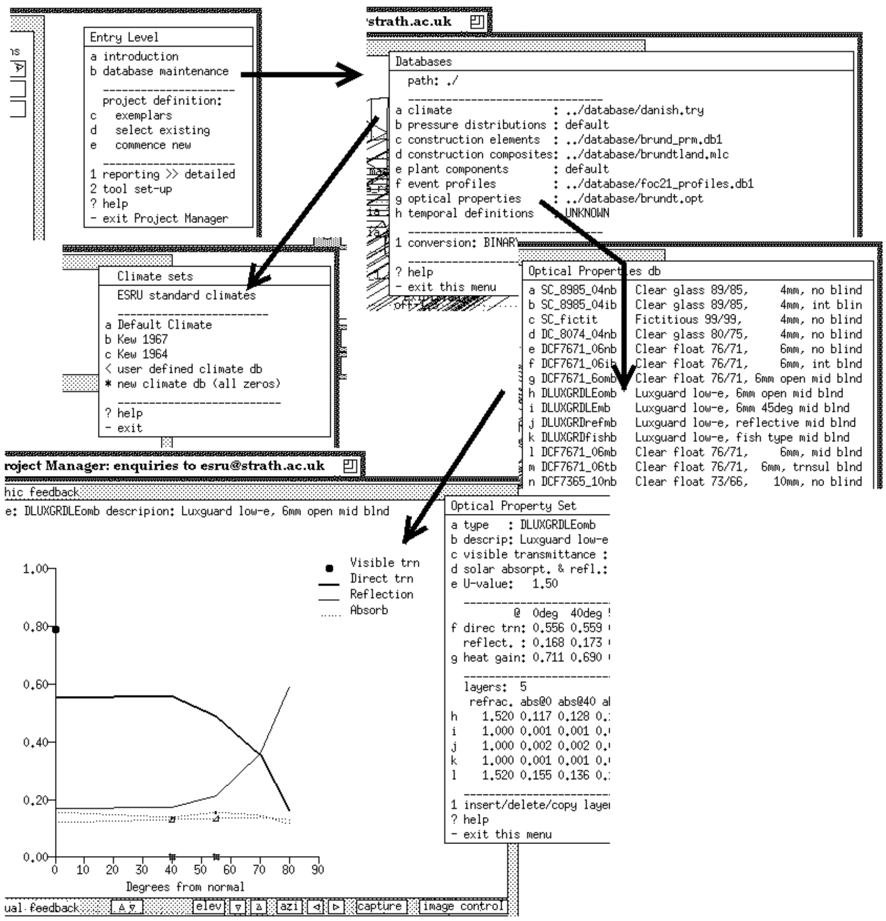


Figure 5.21 Database access facilities.

The materials and constructions databases can be accessed singularly and in combination as shown in Figure 5.22. Materials are grouped into classifications (e.g. metals, glass), each of which has a set of entries. As the number of entries grows, additional classifications can be implemented. The materials list includes data for each entry to aid in scanning for relevant entries. Selection of an entry invokes editing functions for each field. Constructions (e.g. walls, glazing systems, PV modules, etc.) are also accessed via a list, the selection of which allows detailed editing and the selection of materials.

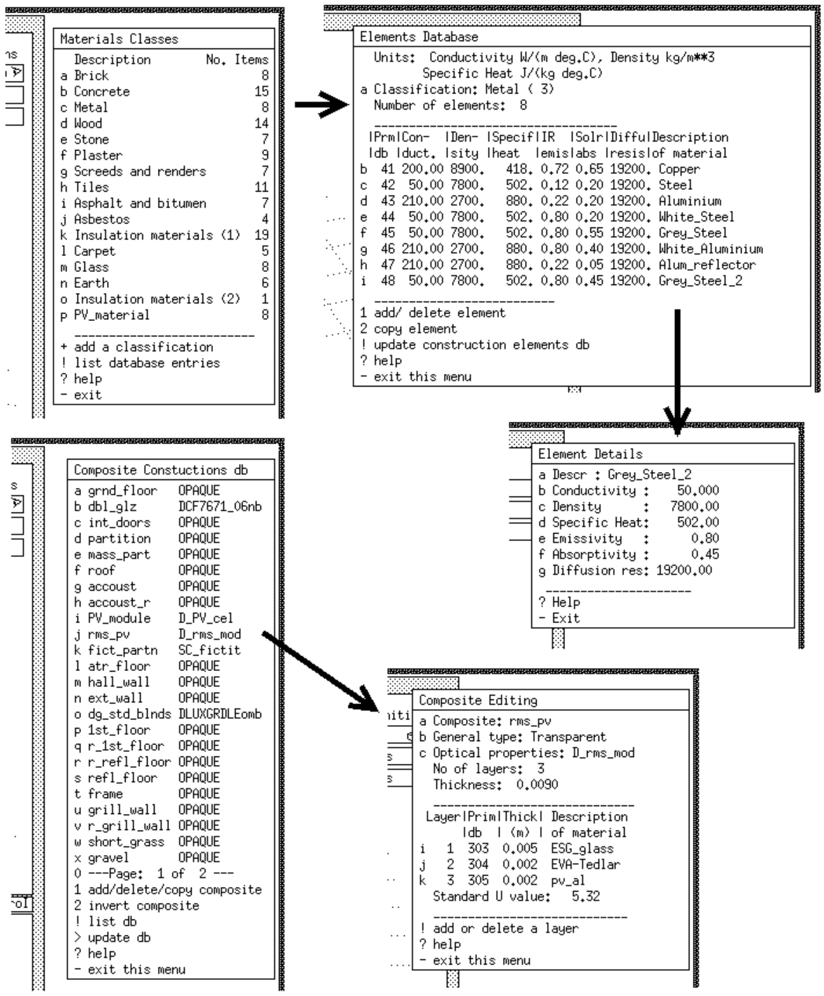


Figure 5.22 Materials and constructions database facilities.

As discussed in Chapter 2, there are a number of dependencies within and between ESP-r's databases. For example, a transparent construction (i.e. glazing) references a named entity in the constructions database and a named optical property set in the optics database. An improved interface would present a single point of definition for fenestration elements which does not require knowledge of the underlying data model.

To ensure that changes in, or corruption to, databases do not unintentionally alter an existing model, ESP-r holds information scanned from the databases in the zone files (defined in Section 2.1). It is these zone files, rather than databases, which are accessed during simulation. Zone files can, of course, be updated as required and the desktop is able to recognise user actions which require such updating. This is but one of a number of consistency issues which are discussed in Section 5.2.5.

After project databases have been established, this information becomes available to support descriptive and computational tasks. For instance, when attributing a surface, its composition is defined by a named pointer to an entry in the constructions database. To alter this attribute, a list of construction database entries is provided along with thermophysical details. In this way the practitioner is offered access to information within each context where such information is required or would reduce ambiguity.

5.2.2 The context of the design

From the opening display shown in Figure 5.23 the user can explore an existing model in terms of these "meta" level concepts. For example, the "Actions" selections (lower portion of the Problem Definition menu in Figure 5.23) allow simulations to be invoked, predictions to be analysed, reports to be generated and the model to be exported to other tools.

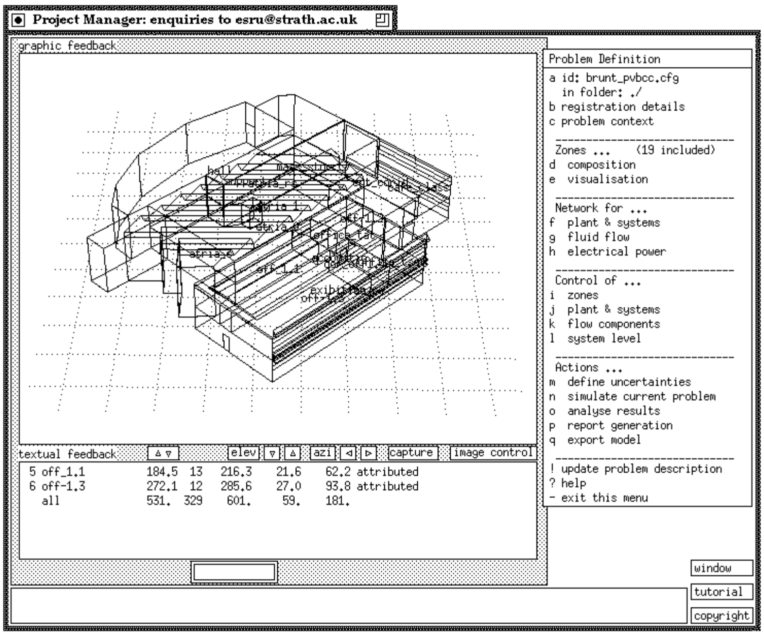


Figure 5.23 Initial Project Manager display after loading a project.

Many users consider the context of the simulation model to be little more than the latitude, longitude and shading from adjacent buildings. The case studies indicated that a broader definition is required. This has been provided by inclusion of a registration facility (Figure 5.9) and expansion of the problem context (Figure 5.24) to include documentation, images, ground topology and macro scale issues such as primary energy conversions and non-zone specific (i.e. dispersed) energy demands.

The inclusion of primary energy conversions allows the atmospheric emissions implications of performance to be explored. Many of the normalised performance indices used for buildings include demands for site lighting, as well as lifts, fans and pumps which may not be represented explicitly in the model. The dispersed demands facility allows such demands to be included, albeit at low resolution.

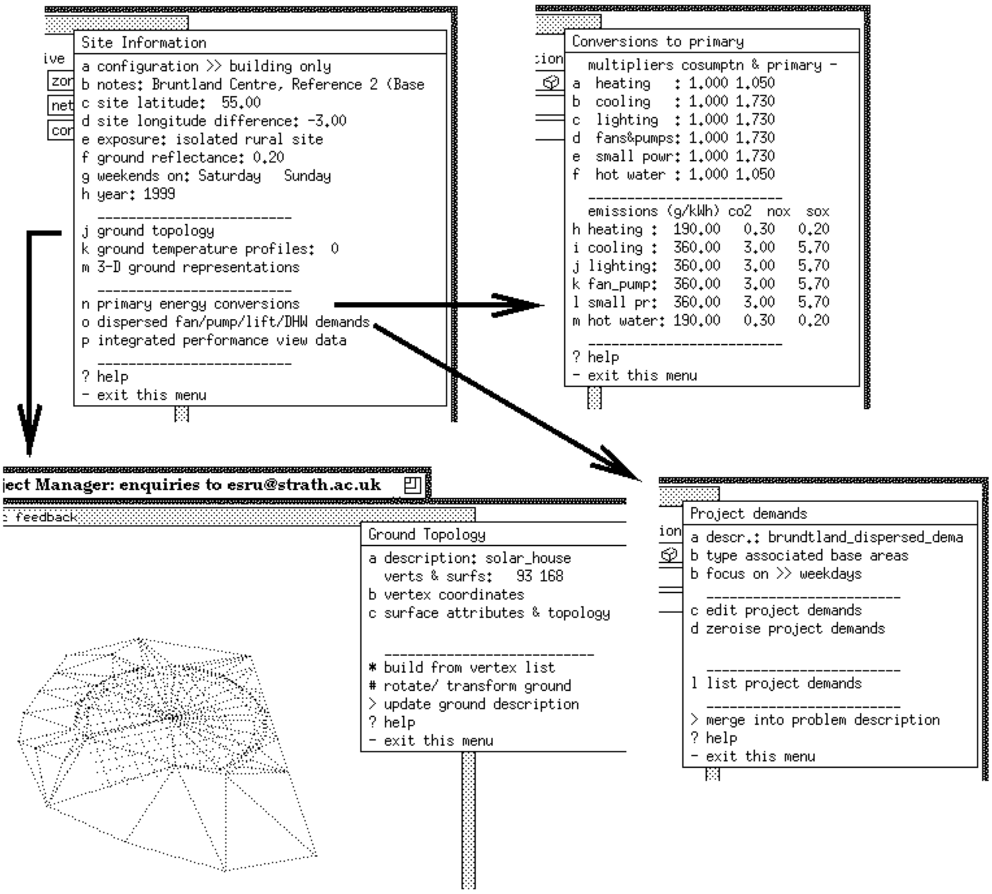


Figure 5.24 Problem context facilities.

Also included in the problem context are facilities to define the focus and metrics of assessment in the form of an Integrated Performance View (IPV). This is an important issue in design decision support which is explored in detail in Chapter 6.

5.2.3 The building definition

The Project Manager is designed to support the ad hoc definition of buildings. Therefore, zones which are not yet complete for purposes of simulation, e.g. are partially attributed or are not fully enclosed, are valid within the Project Manager. The model composition facilities have been designed to make this possible.

When the focus of attention changes from database management to the composition of the model (composition menu choice in Figure 5.23), the menu choices shown in Figure 5.25 become available. In the Problem Definition menu, below registration details and problem context, are facilities for defining zone composition (expanded in right of Figure 5.25), networks for plant systems, fluid flow and electrical power. Below this is the definition of controls for all aspects of the problem and then actions which can be undertaken with respect to the current model.

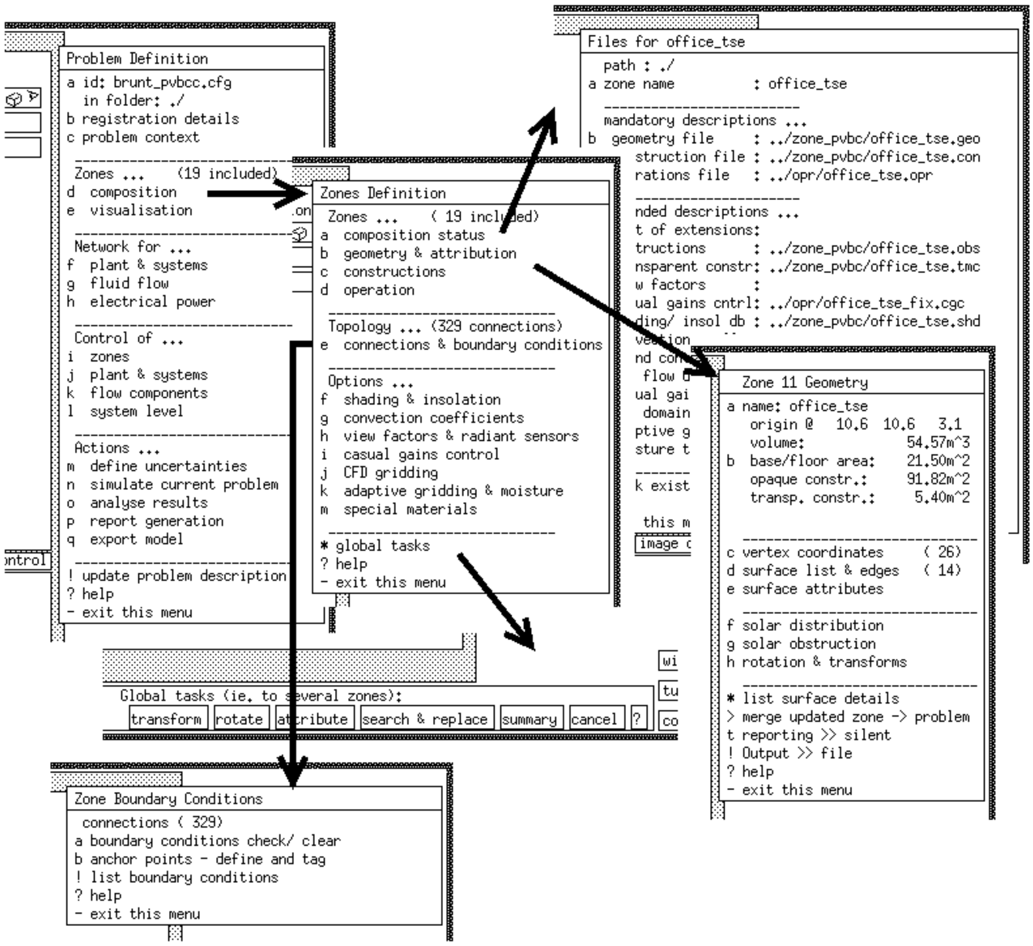


Figure 5.25 Building definition facilities.

Problem Definition

The Problem Definition is the hub of the Project Manager and the placement of functions relative to this point is intentional. Although many practitioners fixate on geometry, it is but one topic among many and it is rightly placed several layers down in the command hierarchy. The placement of the Actions group of options here, rather than the entry level, is in keeping with the tight binding of assessment tasks and current state of a project.

The Zones Definition in the centre of Figure 5.25 changes the focus to aspects of individual zones such as geometry and attribution (note that attribution is accorded equality with geometry), constructions, operations (i.e. required definitions) as well as optional extensions. At the bottom of the Zones Definition are global tasks which act on several zones at one time. These tasks include geometric transforms and rotations, search-and-replace functions (e.g. find all instances of construction single_glz and replace with double_glz) and a summary of zone composition.

In one of the few instances where file names are openly expressed, the "composition status" (expanded at upper right of Figure 5.25) provides a reminder of which facets of the zone have been described.

This is also where experts can manipulate file names, for example, to link all office zones to a common operations definition.

Zone composition and attribution

The lower right of Figure 5.25 shows the zone geometry and attribution facility. Several items are of note:

- the command menu has been overloaded to provide critical feedback;
- vertex coordinates are dealt with separately from the list of surfaces and surface edges;
- definitions related to imposed solar distributions or explicit obstruction blocks are included.

The display and editing of vertices is shown on the left of Figure 5.26. The "Topology of office_tse" menu shown on the right of Figure 5.26 is the focal point for the addition and deletion of surfaces as well as for the editing and checking of edge lists. In both cases, selection of an item results in an editing dialog box providing functions for adding, deleting or copying as well as a set of trigonometric functions. Any alterations are reflected in the graphic display.

The Project Manager also supports basic CAD functionality. For designs dominated by internal partitions, this allows surfaces in adjacent zones (and their attributes) to be easily replicated. Openings in surfaces (typically doors and windows) are handled by an insert function. It requests the opening size and the offsets within the bounding surface and then creates the new surface and redefines the bounding surface to wrap around it. The definition of complex zones is supported by trigonometric functions (bottom left of Figure 5.26) and a point (at vertex) and click graphic input mode.

The in-built CAD functions also support explicit treatment of internal mass as pairs of surfaces within the zone as seen in the book stacks in the library model in Figure 5.27. These are typically formed by defining one face as a surface and then copying and inverting the surface to form the opposite face. Adiabatic boundary conditions are then applied to each and the composition set to a material which is half the depth of the artifact represented (e.g. a 400 mm deep shelf of books might be represented by 200 mm of paper).

Where these geometric manipulation facilities differ from typical CAD application functionality is that the entities are tightly bound to the underlying data structure and topological rules. Moreover, entities are treated as objects, and the implications of user actions (e.g. inserting a door) are distributed throughout the data model in order to maintain consistency.

Once the zone geometry and attribution is complete the data for the zone constructions is conflated from the material, multilayer and optics databases and held in the file store. The use of surface attributes which point to database entries acts as a form of documentation, ensures uniformity in the data model and helps automate the conflation of data into the file store. This said, it is possible to edit specific thermophysical properties. Direct editing of thermophysical properties is, in many respects a

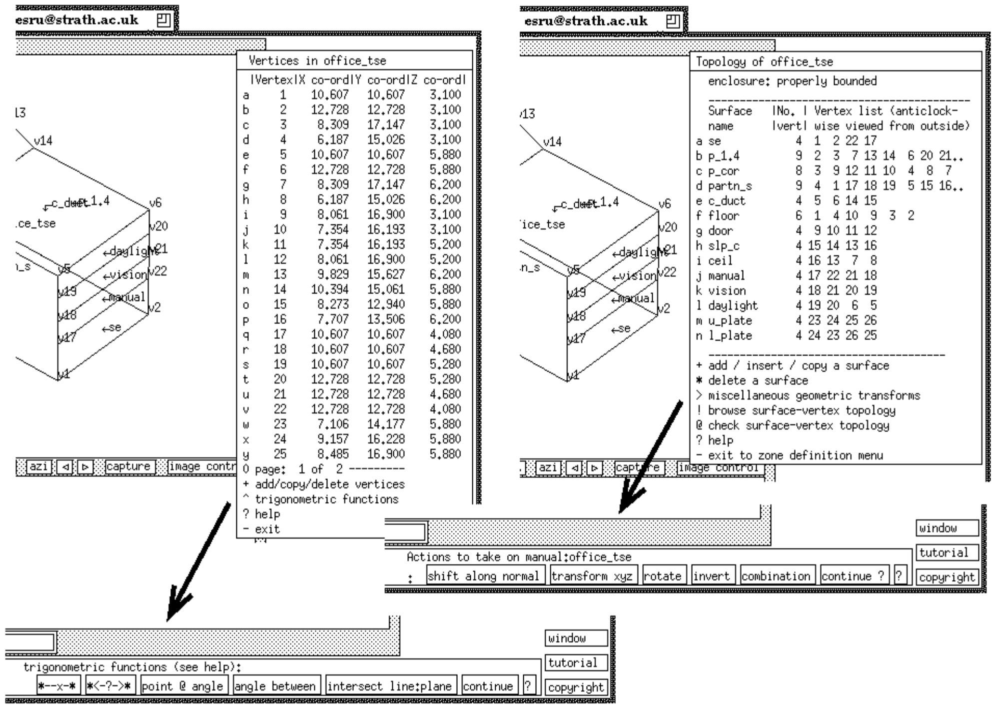


Figure 5.26 Vertex and edge definition facilities.

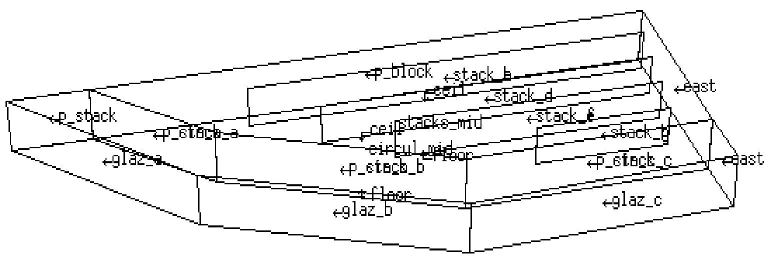


Figure 5.27 Explicit internal mass representations.

historical artifact. It short-circuits the quality assurance, documentation and consistency facilities implicit in the use of surface composition attributes. However, it is a powerful facility in the hands of an expert.

Zone operations

Descriptions of the built environment almost always require representations of occupants and internal gains (e.g. lighting, small power, lifts). Although details vary, simulation tools represent these as schedules or time-step profiles of heat gains with sensible and latent components. For the purposes of engineering estimates or where insufficient information exists for a network or computational fluid dynamics treatment of flow, air infiltration and ventilation between zones may also be represented as schedules or time-step profiles. In ESP-r such operational details are defined on a zone-by-zone basis, although definitions may be shared. The operations facility is shown in Figure 5.28.

The "Zone Operation" menu allows the focus to change between weekdays, Saturdays and Sundays, each of which may hold different air flow and casual gain definitions. The "Air flow" menu defines infiltration and ventilation data for each period. The report in the lower left of the figure relates to the current control applied to infiltration and ventilation. The "Casual Gains" menu specifies casual gains by period. Note that there is an "import from profiles db" command. In the figure this has been used to quickly define occupant and equipment gains.

Such operational schedules map closely to many consultant’s view of building use. Where this is not the case the import of time-step data is also supported by the Project Manager. Where uncertainty in casual gains is a design issue, the work of MacDonald [1998] can be used to determine the effects of parametric variations.

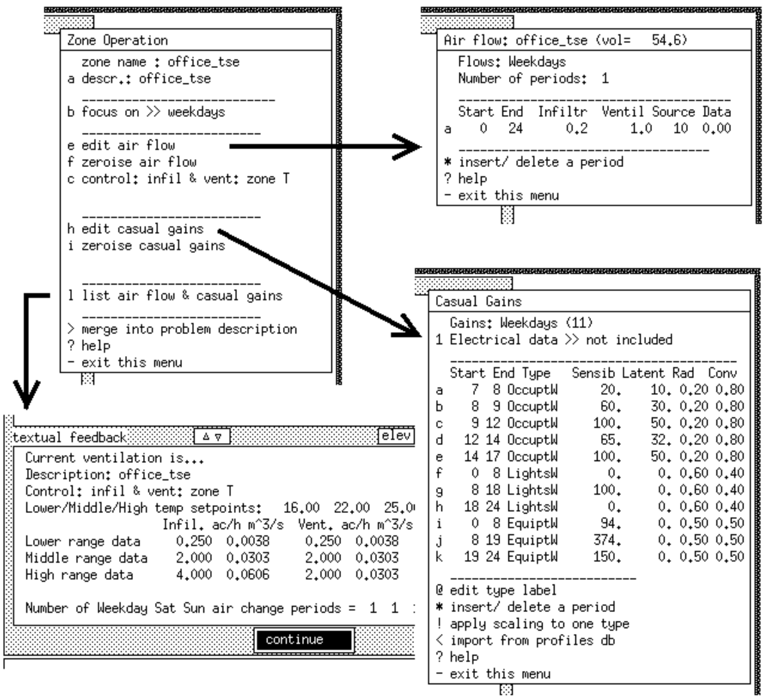


Figure 5.28 Operations (scheduled internal gains and ventilation).

Technical extensions

As stated in the introduction to this chapter, the Project Manager’s design aim was to provide "a structure geared at letting other modules do the work". One way in which this has been developed is in the access provided for technical extensions to models (e.g. shading and insolation, view factors, CFD gridding and adaptive gridding). Such applications are invoked from the desktop (see Figure 5.29) and passed the current problem configuration and, optionally, the current zone of interest. After processing, the revised portion of the data model is incorporated back into the model.

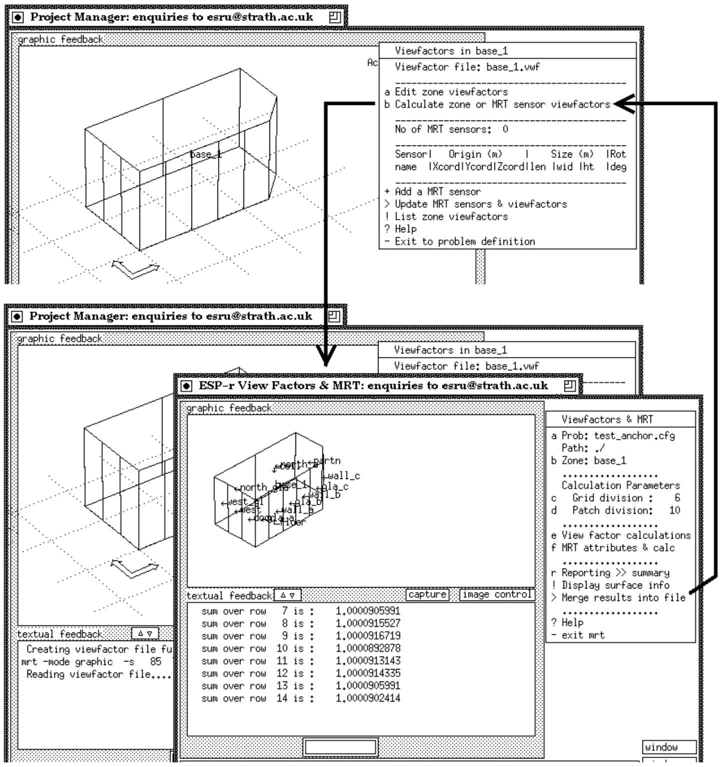


Figure 5.29 Technical extensions via support modules.

The consistency manager is also able to invoke support modules in a "recalculate" mode if required. The structure of the desktop is such that technical extensions need not be ESP-r modules nor do they need to reside on the same computer. In this way tasks can be distributed to make best use of computing and staff resources.

5.2.4 Technical systems definition

ESP-r represents plant equipment, air flow and electrical power as networks of components and linkages. The Project Manager’s treatment of these networks is typified by the air flow network definition facilities shown in Figure 5.30. The high level definition (at the left) accesses separate lists for nodes, components and connections, each of which invokes detailed editing facilities.

Because components (e.g. fans or ducts) are not geometric entities (in the sense that surfaces are polygons in a coordinate system), the definition and understanding of networks is predicated on initial planning, clear documentation and the descriptive names given to entities (e.g. north_west is connected to entrance via extdoor). The network definition facility expects the user to have planned and sketched out the network and defined names for the nodes and components. The author’s experience indicates that a graphic record (see Figure 5.31) is essential for clarity and to support subsequent error checking. With such information the definition process is straightforward.

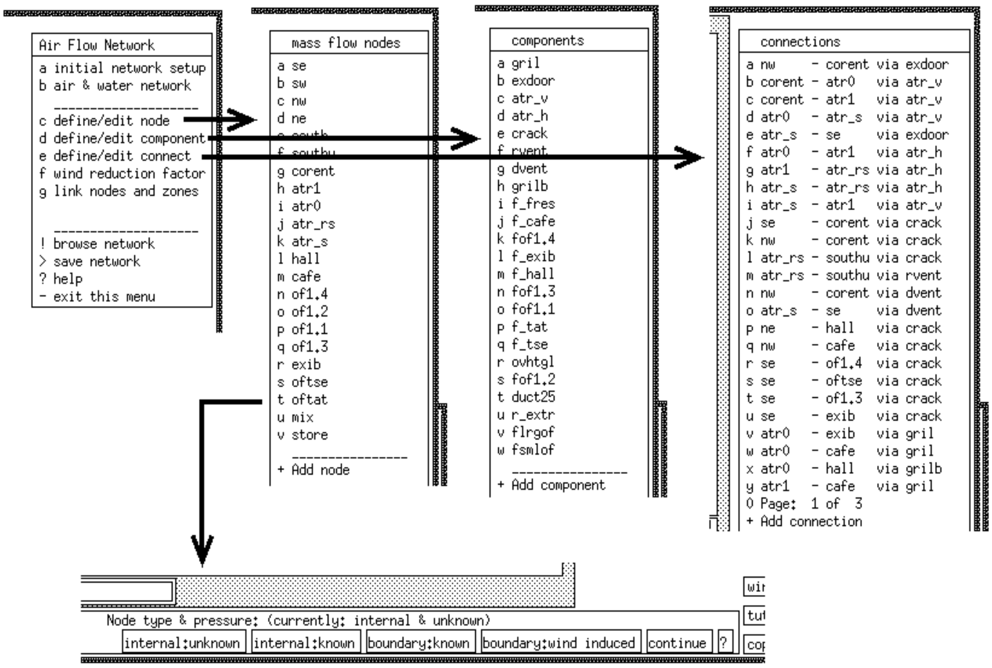


Figure 5.30 Network definition facilities.

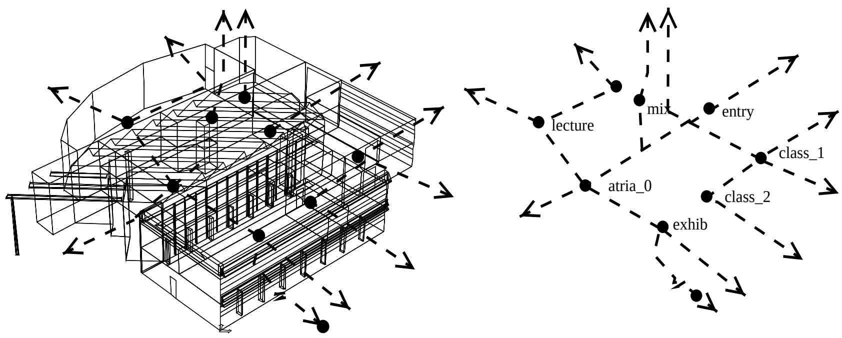


Figure 5.31 Sketch of flow network.

The absence of multiple views (e.g. names, images, graphics) to air, plant and power networks constrains their use. For some users, networks are best understood in a 2D symbolic context such as the icon-based network layout tool shown in Figure 5.32. This tool is invoked from the Project Manager to define the layout of a network and then the data is imported for attribution and incorporation within the model. Further developments are required to fully integrate symbolic definitions into the descriptive process.

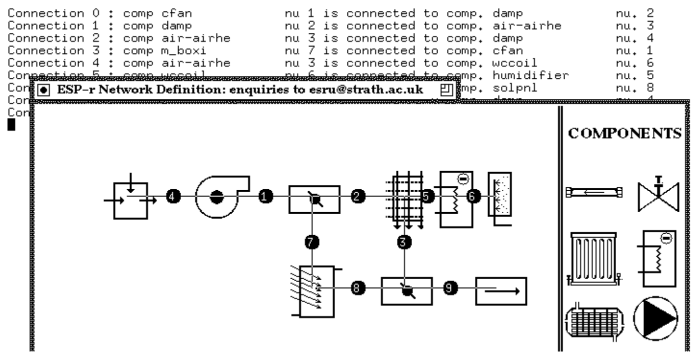


Figure 5.32 Icon based network definitions.

5.2.5 Control definition

Recalling the extent of the control data model and the number of control options presented in Chapter 2, the definition of control is an extensive facility within the Project Manager. The design of ESP-r is predicated on allowing a knowledgeable practitioner to combine sensors (located at virtually any point in the model and sensing almost any condition) with actuators (located at virtually any point in the model) and laws of operation which may change over time.

Control can be applied to zones, network flow and plant. With respect to zones and network flow, the time constant of sensors or actuators is usually not taken into account. Explicit plant control is, on the other hand, capable of representing much of the high-frequency response observed in real control systems. The Project Manager’s control definition facilities focused on zone control are shown in Figure 5.33. High level definitions include documentation, linking control loops to thermal zones and manipulation of the list of control loops. Selecting a control loop opens up facilities for progressively more detailed editing.

The definition of zones, network flow and plant control is dependent on clear definitions of the control regime to be implemented and knowledge of the parameters required. For those who are not control engineers, this process can be problematic and open to mis-use. What is required is a mechanism to indicate the degree of domain knowledge associated with control variants. One approach would be to present the control options in increasing order of complexity and required expertise and provide an overview of which controls are appropriate at different stages of the design process. In addition, those charged with skills acquisition should review the implications of control selection and continue to evolve the exemplars and tutorials related to this topic.

For casual gains (e.g. lighting) and optical and shading devices, control is defined on a zone by zone basis and uses a different metaphor from the sensor/ actuator/ law convention described earlier. Here control is applied to specific entities and is tightly bound to a portion of the data model. The casual gain control facility is shown in Figure 5.34. Each of the data fields or toggles has a direct link to the underlying data structure and sensor positions are indicated within a wireframe view (not included in

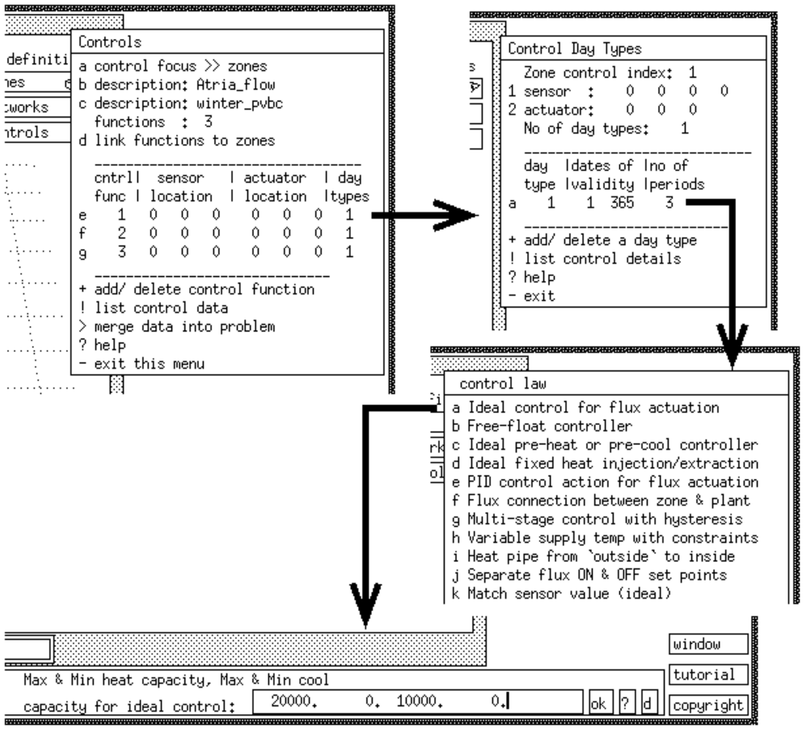


Figure 5.33 Control definition facilities.

the figure). This object-view is supported by context-sensitive and general help, but the number of control variants supported (e.g. radiation sensors, internally derived or user supplied daylight factors, Radiance coupling) is indicative of a facility for use by experienced practitioners rather than novices.

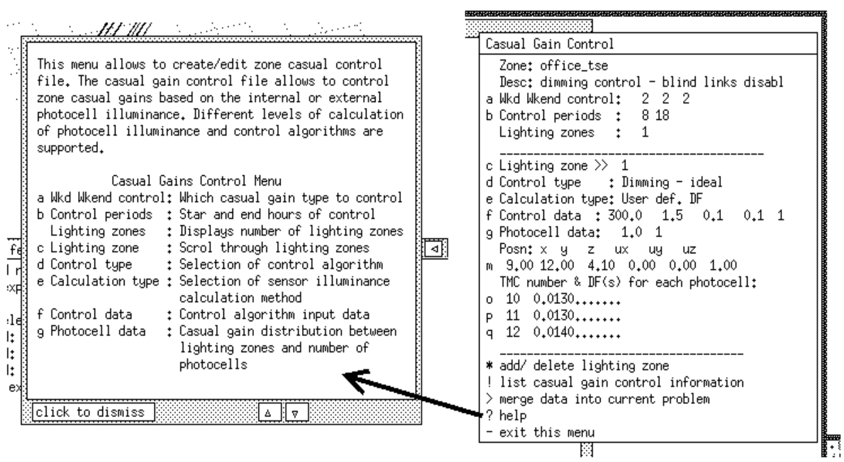


Figure 5.34 Casual gain control facilities.

The use of different control metaphors imposes a cost in terms of skills acquisition, coding resources and interface complexity. Arbitrary conventions such as hour-based control periods in some instances and fractions of an hour in others, or separate treatment of casual gain control are historical artifacts.

There is a case to be made for a similar expression of all controls—if consensus can be reached. The ESP-r data decomposition undertaken as part of the COMBINE project (see Section 7.2) identified a range of treatment of scheduling and control timing within ESP-r. Reconciling these is a considerable task and remains largely out of the scope of the current work.

5.2.6 Ensuring consistency

As discussed in Chapter 2, maintaining consistency within a model involves resolving dependencies within project databases, geometry, surface attributions and composition, controls and networks. It is necessary to resolve the dependencies at (or soon after) each user intervention. For example: that the compositions of two surfaces which form a partition between adjacent zones match; that changes in the definition of a database entry are applied to all referencing items; that when the form of a room changes, view factors and temporal insolation databases are updated.

The resolution of dependencies also involves decisions on the volatility of data. Some dependencies within a model are best resolved at the point of interaction and others are best if delayed. For example, adding or deleting surfaces requires updating both zone level and project level files. While a momentary pause to update geometry and topology is essential, updating viewfactors and shading databases can involve time-consuming calculations. These are better resolved when the user indicates that no further changes are anticipated in that portion of the model. The approach taken is to design a *consistency manager* which flags the dependency type of each user action and, for issues which do not require immediate action, present advice on possible future actions which will resolve dependencies, when the user changes the focus to another part of the model.

When consistency is properly maintained users can work with models of greater complexity and evolve them more quickly. Loss of consistency (i.e. failure to resolve dependencies) is one of the most pernicious of errors and a source of considerable tedium and frustration. User frustration is especially acute when it is not clear who is responsible for consistency and the extent of checking needed. Several techniques are used to maintain consistency:

- interactive editing of objects (which discourages direct file editing and provides feedback on the effect of user actions);
- an edge matching technique to identify possible geometric adjacency;
- logic to detect when volatile information should be saved to file;
- logic to detect when technical extensions (e.g. viewfactors, shading) are out of date.

The attention required to link each surface to an appropriate boundary condition effectively limits models to a few score surfaces unless some assistance is provided. The topology checking procedure, shown in Figure 5.35 is critical in this regard. It finds and matches adjacent surfaces and allows other surfaces to be quickly attributed. As valuable as such assistance is, the practitioner’s understanding of models can be enhanced by a shift from screen based facilities to hard copy reports.

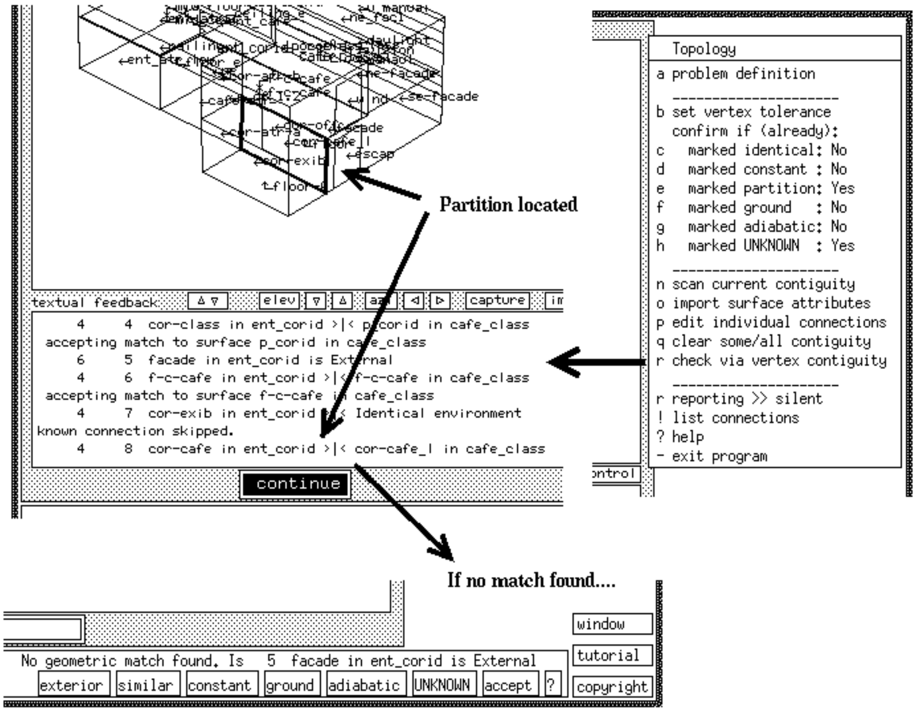


Figure 5.35 Topology checking.

Errors, Understanding and Contiguity

Recalling the observation in Section 4.2, that reporting facilities could be factual and yet not be informative, the design of reports is as critical as it is difficult. Figure 5.36 uses portions of a standard model composition report to demonstrate some of the challenges.

The summary of the model composition report (part a in Figure 5.36) provides a quick overview of the zones in the model, tends to be understood by most users and is often included in consultancy reports. The zone composition report (part b) provides both an overview and critical details on the zone. It is not fully self-explanatory. Many users overlook the "azimuth" column as a critical check that surfaces are facing as intended (in a wire-frame view, surfaces with an azimuth of 90° and 270° can look identical). The detailed zone composition report (part c) is an example of reporting intended for geometric error checking. It presumes knowledge of the data model. The control report (part d) is attempting to express a part of the data model which is only tenuously associated with objects in the user’s domain (only a small proportion of simulation users are control engineers). Understanding of this report requires proficiency in the definition of controls and is dependent on external documentation, such as a sketch of control logic.

Such reporting relies heavily on user supplied names (e.g. surface named railing and cor-atr-prt, compositions named int_doors and fict_partn) which have meaning within the context of the project. Observations show that text-based reporting, regardless of naming strategies, is made less ambiguous if other views of the data model are provided.

```
a: (Model composition report - summary level)
  ID Zone            Volume|          Surface
     Name            m^3   | No. Opaque  Transp  ~Floor
   1 ent_corid        554.8  23   507.6    96.5   135.6 attributed
   2 cafe_class       546.4  20   536.3    47.9   172.7 attributed
   3 cor_off1.4       242.5  22   344.8    10.8    80.7 attributed
   4 off-1.2           74.1   9    99.4    10.8    24.9 part-attributed
   . . .

b: (Zone composition report - summary level)
   Zone ent_corid ( 1) is composed of 23 surfaces and 51 vertices.
   It encloses a volume of 554.8m3 of space, with a total opaque surface
   area of 604.08m2 & approx floor area of 135.57m2

   A summary of the surfaces in ent_corid( 1) follows:
   Sur|    Areas    |Azim|Elev| surface    |geometry| multilayer  |environment
      |Gross | Net  |deg |deg | name       |type|loc| constr name |other side
    1    5.58   5.58 225.   0. cor-off-prt  OPAQ VERT int_doors    SIMILAR
    2    9.30   9.30 315.   0. cor-atr-prt  TRAN VERT fict_partn   atria_1
   . . .

c: (Zone composition report - detail level)
   The zone is defined by the following general polygon vertices:
   Vertex|X-coord.| Y-coord.| Z-coord.|Associated surfaces     |
      1   -16.971    21.213     0.000   1,  8,  9,
      2   -19.092    23.335     0.000   1,  2,  9,
    . . .
   Each surface (polygon) is composed of vertices as follows:
   Surface|No.of   | vertex list (anticlockwise viewed from ext)
         |Vertices|
       1        6    2,  1,  9, 21, 22, 10,
       2        6    3,  2, 10, 22, 23, 11,
   . . .

d: (Control report - detail level)
  Flow control: chanl_to_off :17 loops.

  The sensor for function 1 measures mass flow node:     atr0
  The actuator for function  1 is mass flow component:   2 exdoor

   Day type  1 is valid period: Fri  1 Jan to Fri 31 Dec, 1999
   and contains  3 control periods.
   Period| Start |Sensed   |Actuated | Control law | Set
   no    | time  |property |property |             |point
      1    0.00   dry bulb > flow       on  off    100.0     1.0
      2    8.00   dry bulb > flow       on  off      5.0     1.0
      3   19.00   dry bulb > flow       on  off    100.0     1.0
```

Figure 5.36 Model composition report.

5.3 Integrated Performance Views

> *When we reason about quantitative evidence, certain methods for displaying and analyzing data are better than others. Superior methods are more likely to produce truthful, credible, and precise findings. The difference between an excellent analysis and a faulty one can sometimes have momentous consequences [Tufte 1997 p27].*

Thus far the discussion has focused on the creation of models. However, the confirmation of initial design conjectures and the recognition of unintended consequences of design decisions requires access to performance data in ways which make clear the initial metrics of the design project. Simulation has traditionally provided access to performance predictions in a number of tabular and graphic forms and within standard reporting and data export functions such as shown in Figure 5.37. However, the task of gathering performance data and applying the metrics of the current project has largely been left to the practitioner and potentially important causal relationships may go unnoticed. For example, short period assessments are an excellent vehicle for understanding performance because the data set is small enough to be absorbed and the underlying causal patterns understood. Yet Donn [1997] found that 55% of US respondents to a simulation use survey never ran simulations for typical days.

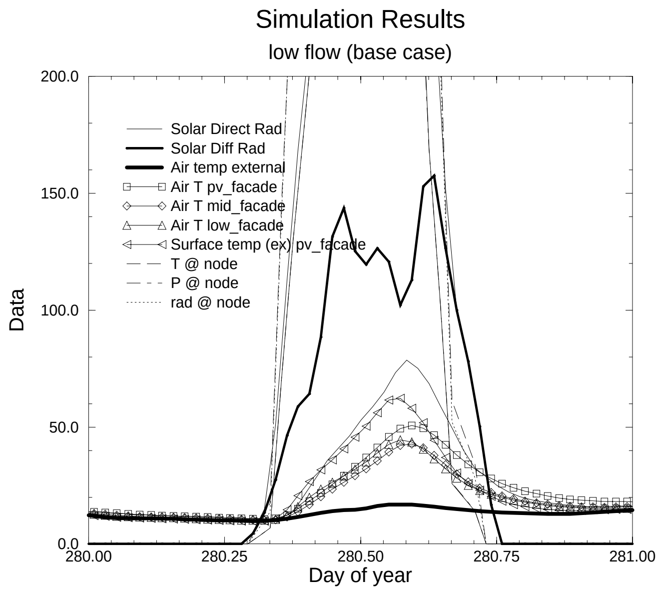


Figure 5.37 Report quality performance graphs (exported).

There are also increasing demands in European research programmes to ensure that new technology is assessed against a broad range of criteria. Where programmes are carried out by a number of partners, it can be both tedious and expensive to conflate the many sources of information and ensure that the underlying methods and metrics used by the various teams are compatible. In coming to an agreement on the metrics to be used and the information needed to support multi-criteria assessments in the Solar House, Image and DL-E programmes, a consensus on the form of an Integrated Performance View (IPV) [Clarke et al. 1996] began to take shape and this has since been further refined [Clarke, Hand and Janak 1998].

An IPV is a considered and compact view of a set of performance indices which supports the unambiguous comparison of design options (via printing on transparency material or by rapid switching of images on a monitor). An IPV would include annual, seasonal and daily performance data as well as a synopsis of the design variant and model. IPVs differ from conventional reporting schemes in that their definition begins as the model is being created and evolves with the model as in Figure 5.38.

Define base case and reference simulation models and begin IPV documentation.

↓

Define IPV in terms of: a) periods of seasonal assessments and typical days, b) scaling of model (in part or whole) to whole building, c) identify areas of model which relate to specific issues (thermal/visual comfort, component performance), d) identify performance metrics.

↓

Undertake sequence(s) of simulations and generate IPV report as per IPV definition.

↓

Produce IPV from the report and use visual and /or numerical comparisons to identify issues. If necessary, confirm trends by looking at detailed performance.

Figure 5.38 IPV decisions and tasks.

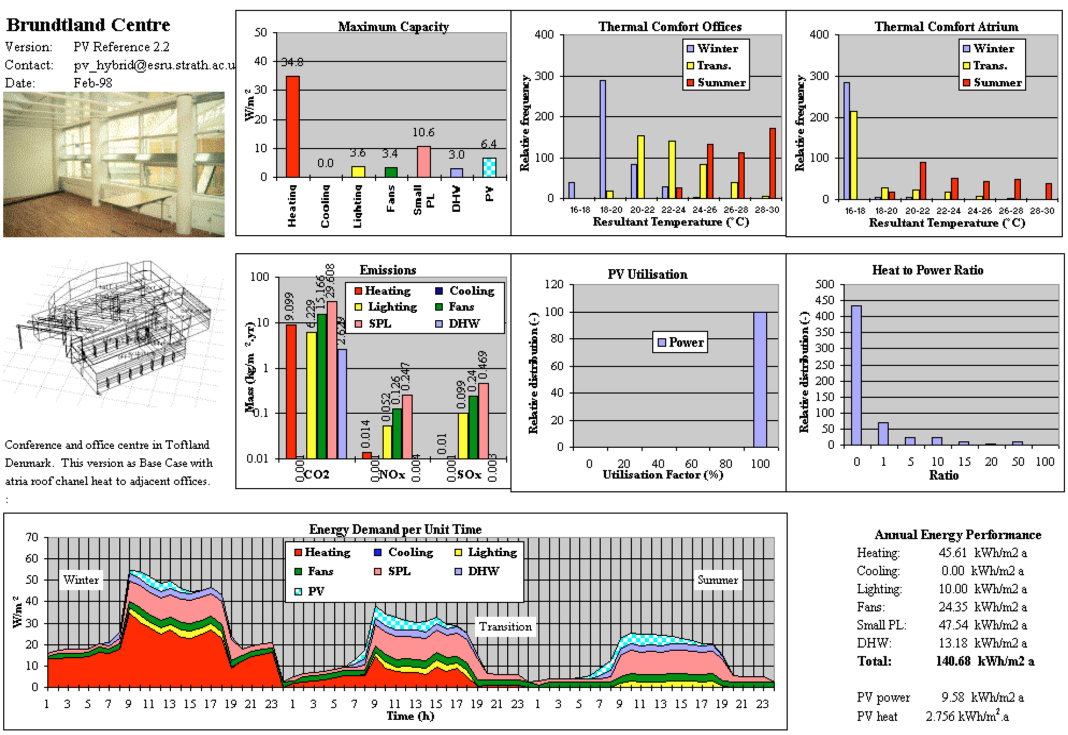


Figure 5.39 Brundtland Centre Reference case IPV.

Figure 5.39 shows an IPV for an Exhibition Centre in Toftlund, Denmark. Taking each portion of the IPV in turn, the upper left identifies the design variant and provides a synopsis of its features. The three graphs across the top deal with capacity (W/m^2) and thermal comfort (relative frequency of resultant temperatures) in offices and the atrium. The middle row of graphs deals with environmental emissions ($kg/m^2.\ yr$) at source and the metrics of PV performance (power utilisation and heat-to-power ratios). The lower right of the IPV lists annual performance indices. To complement these integrated values, typical daily patterns of energy demand for each season are included in the bottom graph. Each of the energy-related graphs includes data for aggregate heating, cooling, lighting, fans, small power, domestic hot water and, in this case, PV performance.

IPVs call attention to differences in performance and the overall implications of design decisions. In addition to visual comparisons, the underlying performance data can be used to guide detailed explorations for causal effects. A first step might be a numerical comparison of performance data such as is shown in Figure 5.40 and then the use of causal chaining of energy balances (described in Figure 4.1).

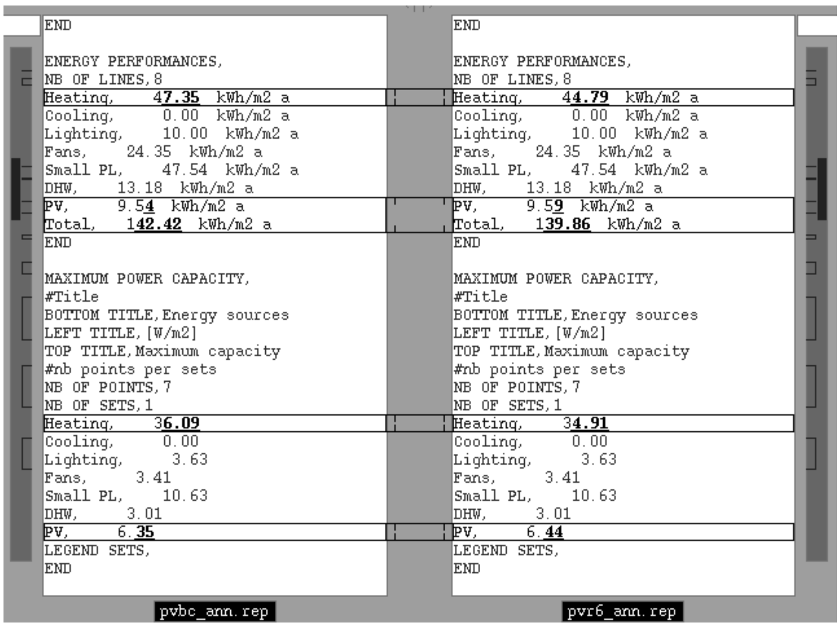


Figure 5.40 Scanning for numerical differences.

IPVs have been used as a core reporting mechanism within a number of research programmes and consulting projects [Clarke et al. 1997]. The generic nature of the IPV has been confirmed by its use of data from TSBI3 [Johnsen and Grau 1994] and APACHE [Facet 1995] as well as ESP-r. Several charting environments have been used to produce IPVs. IPVs are a significant realisation of the initial specification and are the result of incremental refinements over three years. The author’s contribution has been to:

a) develop a formal descriptive language to define what is included;

b) develop a neutral reporting format to hold performance data to be presented;

c) extend the simulation product model to support issues of atmospheric emissions, energy demands which are not zone specific, and seasonal and typical day assessment periods;

d) adapt the results analysis module to take the formal definition and extract, conflate and scale the various data sources into the neutral reporting format for presentation.

The need for a formal description follows from observations of the degree of commonality within projects, the need to support exceptions (in design focus) as well as the potential to improve quality assurance. The formal description acknowledges that the metrics of the project and decisions on what performance aspects need to be included in the IPV are often defined early in a project and should become part of the simulation model (see Table 5.3).

The scaling factors in Table 5.3 might be ratios of time (in the case of small power), ratios of heating and cooling degree days (for environmental systems) or luminance (for light switching). There are separate descriptions for thermal and visual comfort, each defining focus zones and what metric (resultant temperature or PPD) to use. Energy capacity, annual use and typical daily performance are defined so that some zones may be treated as, say, typical offices to be scaled, while other zones are used explicitly.

Table 5.3 IPV definition.

Tag	Data	(Description)
* IPV		(file identifier)
*title	Brundtland Centre	(main title)
*version	Image Reference 3	(model variant)
*contact	pv_hybrid@esru.strath.ac.uk	(project contact)
*date	January 98	(project date)
*syn1	Exhibition centre in Toftlund	(synopsis)
*syn2	Denmark. This version as the Base Case	
*syn3	with upper atrium air heating to DHW.	
*periods	4 11 96 103 200 207	(winter, transition, summer)
*typical_days	6 99 204	(days for hourly graphs)
season multipliers:	heating|cooling|lights|fans|smallp|DWH	
*win	12.07 1.00 15.200 15.200 15.200 15.200	(winter scaling factors)
*trn	12.57 1.00 15.200 15.200 15.200 15.200	(transition scaling factors)
*sum	15.72 1.00 15.200 15.200 15.200 15.200	(summer scaling factors)
*therm_comfort	7 1 588.0	(thermal comfort focus data)
	3 4 5 6 7 11 13	(list of zones)
*visual_comfort	1 0 18.7	(visual comfort focus data)
	11	(list of zones)
*energy_1	15 1560.0 1.000	(energy data, first set)
	1 2 3 4 5 6 7 8 9 10 14 16 17 18 19	(list of zones)
*energy_2	0 1. 1.	(energy data, second set)
*dmds	0 0 0 1 0 1 1	(distributed energy demands)
*end		

Two general forms of IPV report have been tested. One included only raw performance data and relied on the display application to provide conflation and scaling. The second included only presentation data (i.e. no computations were required in the display application). The latter allowed IPV reports to be numerically as well as visually compared.

A number of the performance metrics within the IPV proved to be outwith the simulation product model. For example, lifts and domestic hot water are a significant energy demand in many buildings, but do not often feature in thermal simulations. Assessing the CO_2 implications of demand-side management decisions requires, at a minimum, a conversion between the various energy demands within a project and primary power generation.

For IPVs to be regularly and reliably used, it is necessary that their production become unremarkable and involve a minimum of "hand-crafting". With the advent of a formal IPV definition, and extension of the data model, it is possible to automate IPV report generation. The results analysis module of ESP-r has been extended to extract the relevant performance data and to conflate, scale and export the results. A further discussion of IPVs is included in Chapter 6.

5.4 Combining Thermal and Visual Assessments

Although visual assessments have often been associated with high resolution images, it is the engineering data needed to drive lighting and blind controls and find the relative frequency of visual discomfort which has been critically absent from design decision support. Degelman [1998] notes that the energy savings associated with daylighting have not been fully exploited and asks for "...software tools capable of simulating responses of lighting controls to the availability of daylight on a short time interval." A system which achieves this, based on the cooperative use of ESP-r and Radiance [Clarke, Janak and Ruyssevelt 1998, Janak 1998], is presented in this section. It demonstrates many aspects of the problem of exchanging data models and imposing control on a "peer" application to ensure that tasks (creating, manipulating the visual model, undertaking standard assessments) are carried out. A typical visualisation session is shown in Figure 5.41.

Radiance [Ward Larson and Shakespeare 1998] is a visual simulation tool which is capable of providing numerical output as well as images approaching photographic quality. There is a considerable degree of similarity between the Radiance and ESP-r data models, although Radiance entities lack many of the topological constraints associated with thermal simulation and the format of the file store is minimally defined. Translation is a straightforward process, at least from ESP-r to Radiance (the lack of structure makes a reverse transform difficult).

What complicates matters and suggests that something more than a filter is required, is that Radiance is composed of two dozen modules, each of which has a dozen, often sensitive, control parameters. The user has considerable freedom in how to employ these modules and their control parameters to achieve a desired result. Stated less sympathetically, there is essentially no interface to Radiance and much syntactic knowledge is associated with its use. If simulation is noted for a steep learning curve, lighting simulation is, for some users, a black hole.

There are a number of visual assessments (e.g. patterns of shading, daylight distributions and glare assessments) which would provide useful feedback to the design process if such tasks could be carried

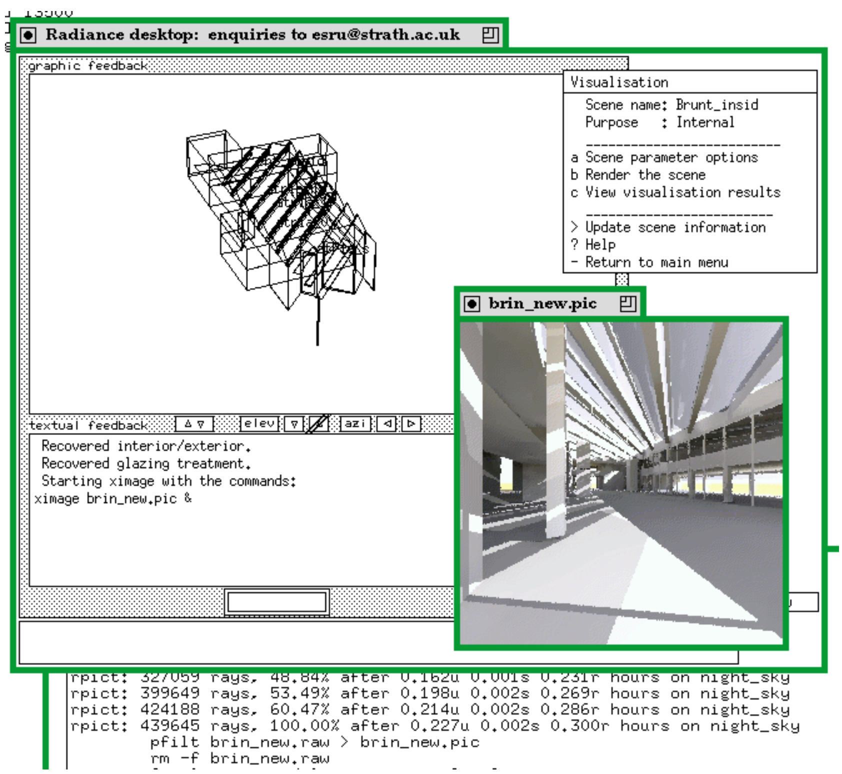


Figure 5.41 Visualisation session.

out without the need to edit files manually, remember combinations of parameters and control each Radiance module. Although there are visual assessments which demand considerable expertise, it should, for example, be possible for a novice to view patterns of shading on the outside of a thermal model.

What is required is to isolate the user, whether novice or expert, from the mechanistic complexity of visual assessment, impose order and attribution on the visual model and *drive* the modules to accomplish specific tasks. Given the success of the Project Manager in this regard, the approach taken by the author, with the assistance of Dr. Milan Janak and Iain MacDonald, has been to implement a tool which both creates the descriptive entities of Radiance and drives the Radiance suite.

This tool uses the same desktop and attribution metaphors for visual assessments as the Project Manager. It owns the visual description, imposes order and internal documentation on the native Radiance descriptive files in support of particular assessment tasks and for quality assurance. Those who have little or no knowledge of visual simulation are able to create a default image of the current ESP-r model with a few dozen keystrokes. The visualisation session in Figure 5.41 shows its use.

The tool supports the creation of "scenes" (e.g. for external views, internal views, daylight factors), each of which comprises a visual model, related computational parameters and view points. It translates the thermal model into equivalent visual entities, executes Radiance modules to create an internal representation and invokes image and/ or numerical calculations as required. The author of Radiance described the files created by this ESP-r tool as the most highly attributed and readable of any translation he had seen [Ward Larsen 1996].

For those with knowledge of Radiance there is an interactive mode (see Figure 5.42) where view points, descriptive files and calculation parameters can be defined. The definition of viewpoints, shown in the middle right of the Figure 5.42, provides an alternative to the use of the Radiance rendering tools to set initial viewpoints. The lower left menu allows visual simulation parameters to be set and converted into appropriate commands depending on the nature of the visual assessment task.

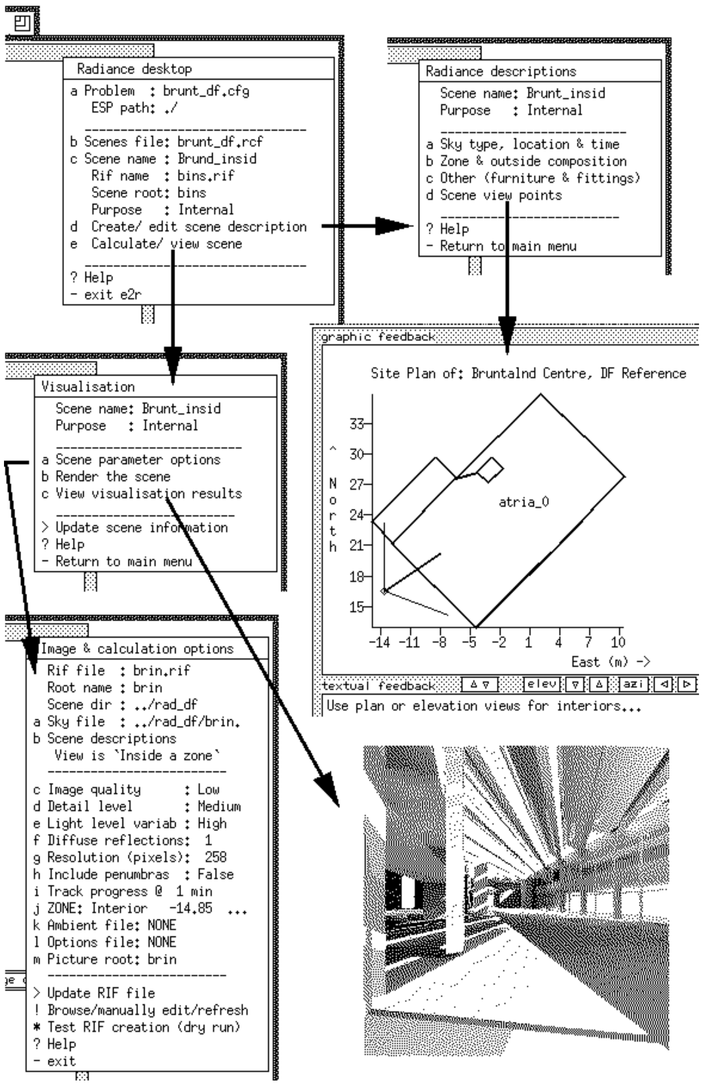


Figure 5.42 E2r command structure.

Depending on the nature of the visual assessment, it may be necessary to include additional geometric detail (such as framing details) to the visual model created from the thermal model. A number of approaches to the creation and maintenance of thermal and visual models have been tested by the author and other ESRU staff. The consensus reached is that design decision support in evolving projects is best served by overloading the thermal model with non-thermal objects (e.g. obstruction blocks, ground topology) as shown in Figure 5.43. Here, the colour rendering model is equivalent to the thermal model and this level of detail is sufficient to determine daylight factor distributions, glare sources (as perceived by occupants in the atrium), sunlight distributions and the like.

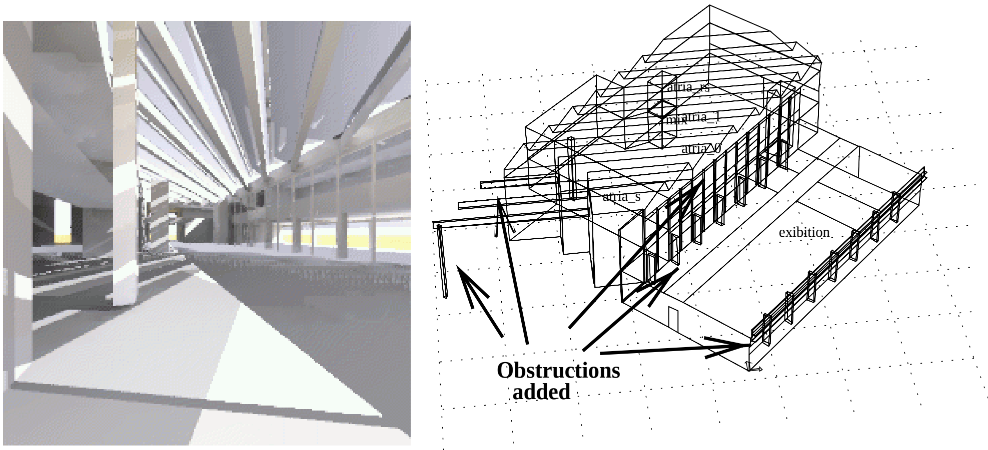


Figure 5.43 Thermal model sources for daylighting issues.

The tool is also useful in the generation of visual performance indicators (e.g. glare, daylight factors, visual comfort). In Figure 5.44, six displays are included. Proceeding clockwise from the upper centre: glare is assessed from viewpoints parallel and perpendicular to the atrium facade of an office (glare is worse looking parallel to the glass); daylight factor plots as a function of distance from the glass; visual comfort as predicted by the Guth Probability index [CIE 1983]; visual comfort as predicted by the JPPD index [Compagnon 1997] and, in the upper left the source image from which the JPPD index was derived.

Such data can be used as static input to control switching in a thermal simulation. Alternatively, light levels at sensor points can be re-assessed as the thermal simulation progresses [Janak 1998]. This is enabled by the integrated simulator invoking a visual simulation at each time step, passing it the current weather data and sun position and taking back data to pass to a photocell defined as part of the lighting control system. Janak also reports on the development of a daylight distribution database, where the contribution of 144 sky patches is calculated for each sensor point. This method supports extended short timestep assessments of lighting (e.g. annual assessments at 1 minute timesteps are possible). Additional discussion of cooperative tool use and integrated performance assessments may be found in Chapter 6.

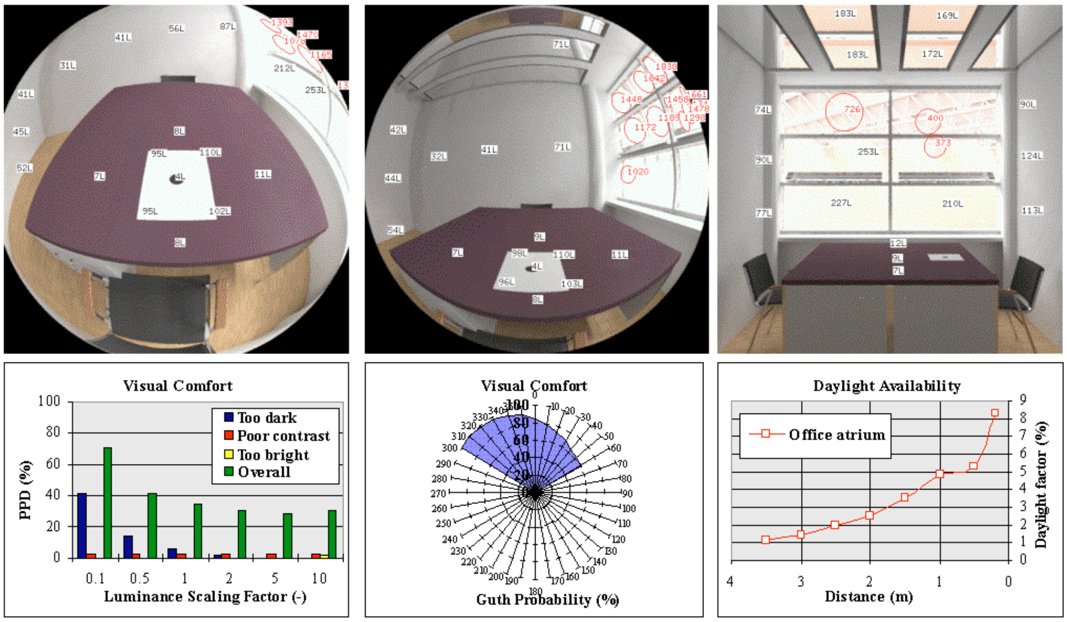


Figure 5.44 Reporting of daylighting issues.

5.5 Corporate use of simulation

The Solar House [Fitzgerald and Lewis 1996] and Daylight Europe [Clarke et al. 1996] projects included scores of designs, each involving different goals and assessment techniques at different stages of completion. Such a mix of projects is not uncommon in professional firms and thus it is beneficial to provide a central data store which supports access to information, irrespective of the project focus and degree of completion.

The Project Manager has been designed so that knowledge of the name and location of the project configuration is the sole requirement for access to all aspects of a project. This has been used to support access to exemplars. The benefits of opportunistic browsing of past models extend beyond training exercises to the deployment of simulation within professional practice. It allows a firm to encapsulate prior projects so that they can be used to support demonstrations to new clients or be accessed in the planning stage of a new project to explore possible appraisal strategies. Early evidence on evolving working practices from firms which have adopted ESP-r and have undergone mentor based training is encouraging. Among other things, desktop access to projects allows:

- models to be arranged hierarchically by project and accessible for high level and detailed browsing;
- on-line or off-line use, with or without geometric information;
- data exchange with other applications including CAD;
- selective enhancement of project description (environmental systems, lighting, control, etc.) as the design evolves.

Implementing access to scores of projects for the purposes of maintaining a corporate project database (the selection list from the Solar House programme is shown in Figure 5.45), or to access exemplars for training purposes, requires the management of a list of project configurations. A portion of a project's database is shown in Figure 5.46. The corporate use of exemplar mechanisms is discussed in Appendix D.

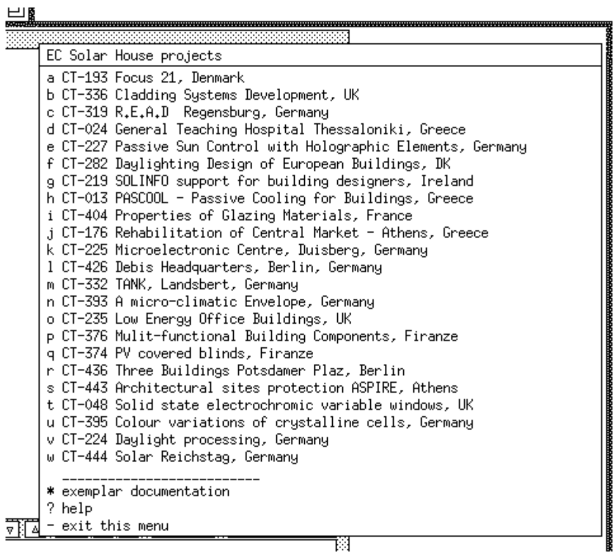


Figure 5.45 Corporate database of projects.

```
*EXEMPLARS
*help   Energy Systems Research Unit projects database. Solar
*help   House group is read-only (see admin to update data-store).
*group  EC Solar House projects
*help   A collection of 35 projects comprising the EC's Solar
*help   House Joule II programme (Nov 1992 - December 1996).
*help   The programme includes 18 low energy building designs,
*help   11 component developments, 5 dissemination projects
*help   and 1 urban-scale development.
*item
*name   CT-193 Focus 21, Denmark
*aide   The Brundtland Centre: a low energy exhibition centre
*aide   in Toftlund, Denmark.
*cfg    /usr/esru/Projects/Solar_h/CT193/cfg/brund.cfg
*root   /usr/esru/Projects/Solar_h/CT193/cfg
*item
*name   CT-319 R.E.A.D  Regensburg, Germany
*aide   Urban energy conservation study, Regensburg, Germany
*cfg    /usr/esru/Projects/Solar_h/CT319/regen-g2g.cfg
*root   /usr/esru/Projects/Solar_h/CT319
  . . .
```

Figure 5.46 Projects database.

The corporate use of simulation also implies the creation and maintenance of corporate databases, the observation of work in progress, analysing successful and unsuccessful working patterns and distributing tasks within the group. The Project Manager can be configured so that specific sets of corporate databases, climate sets and supporting tools are associated with different users. All projects can be placed on-line and managers and staff can access them as required.

The audit trail facility described in Section 5.1.6 supports an understanding of how simulation is used. This in turn supports quality assurance by ensuring that, for example, topology has been checked. From the viewpoint of the project manager, the audit trail of the quality assurance offices can also be reviewed.

5.6 How far have the barriers shifted?

The preceding sections have described the design, organisation and facilities of the Project Manager. The desktop has been installed in a number of settings and its evolving use observed over several years. These observations have provided indications as to the efficacy of the approach in terms of design decision support. Changes in the rate of skills acquisition, the nature and complexity of simulation models and in simulation’s support for design are summarised in the following paragraphs.

The use of interface conventions, such as the juxtaposition of named attributes with graphic entities, allows the complexity of models to increase without an increase in the overall resource required to mount a simulation exercise. It was also found that additional geometric feedback and model documentation reduced errors and enhanced understanding, but depended on users being willing to adapt their work practices. Ways must still be found to make clear descriptions such as "tile floor in cold storage locker of the Abram’s complex, advanced glazing design variant", preferable to "surface 4 in zone 6 of project agp3b".

The pragmatic nature of the interface imposed some limitations on simulation work and the information that the practitioner is able to recover from the use of the tool: the lack of colour limits the clarity of graphs; the wire-frame refresh/redraw cycle can be a distraction and sometimes symbols can be more concise than words.

One initial conjecture, that alternative descriptive facilities would benefit users, has not been proven. A case in point: thermophysical attributes of surfaces can be defined either in the zone composition or zone constructions facility. Observations showed that the existence of substantially similar facilities led users to assume that different information was, in fact, being requested. Thus, an approach which expresses all information related to an entity in one location, appears to have considerable benefits.

One goal of the desktop was to allow users to shift their attention away from tool interactions to higher (methodological) aspects of simulation practice. While the evidence thus far is encouraging for most types of users, novices continue to mistake mastery of interaction as an end-goal rather than something that enables them to use the underlying functionality to explore ideas. Novices display a disconcerting tendency to compose their models from the keyboard. What emerges from such impromptu sessions is invariably flawed and reinforces the observation that pencil, paper and planning sketches are still among the most powerful tools for simulation.

Corporate use of simulation requires the observation of work in progress, analysing successful and unsuccessful working patterns and distributing tasks within the group. The in-built audit trail is a valuable tool for such tasks, but suffers from being a passive device.

When the on-screen tutorial was developed, the initial idea was to complement the printed manual. In practice it also complemented the context help for issues such as environmental systems control which could not easily be condensed. Users would typically begin by browsing topics, returning to selected items for further (often intense) inspection. Some users kept the tutorial on screen throughout their work session, some used it for specific enquiries and others made little use of it. It proved difficult to satisfy both the student (who felt the facility inadequate because it did not, for example, offer a basic discourse on the term "clo value") and the control systems engineer (who, for example, found the text supporting fuzzy logic controls to be academic).

The tutorial module was an incremental advance and a useful adjunct to user manuals and context-sensitive help associated with user dialogue. It provided a foundation for further development, and much of what was learned was re-expressed and extended in one of the first WWW based simulation tutorials. Over time, the depth and breadth of topics expanded and new uses were found for hypertext facilities and others have built on the initial foundation with some success. Mumaw [1997] has been instrumental in merging the workshop exercises into the body of the hypertext and expanding its scope beyond a simulation tool tutorial to present both an introduction to simulation as well as detailed explorations into the underlying computations <http://www.strath.ac.uk/Departments/ESRU/>. A TEMPUS project [Hensen 1998] has the objective of supporting simulation based instruction in Eastern Europe. By mid 1998 ESRU pages were being accessed from scores of countries, albeit that the University of Strathclyde is the primary source of requests when simulation courses are running.

The many benefits of hypertext must be considered against its limitations. Even compared with high resolution monitors, the printed page is able to present information at greater density and resolution. Some in the simulation community would prefer not to lose (or sacrifice the quality of) hard-copy reference materials and users' guides.

The goal of supporting the transparent transfer of simulation models and providing access to well documented and well structured example problems (exemplars) has largely been achieved. It is currently an unremarkable event for a consultant in Kuwait to ask for guidance on some aspect of an office building and ESRU staff to receive and review the model, commission a short assessment and be discussing how to proceed within the space of a half hour. Participants in the first afternoon of a training workshop are able to commission a series of basic simulations on a model of which they have had no prior experience. Design firms can have on-line access to scores of past projects (see Figure 5.1.7).

Other patterns have emerged from observations of the Project Manager in a number of contexts.

- A new generation has emerged of competent, but not expert users who have little or no concept of, or interest in, the underlying file structure. Experts and archivists are, of course free to pursue their traditional activities.
- Training is much less concerned with keyboard skills, navigation and the interpretation of models. Simple models can be composed and assessed without knowledge of application names or the nature of the data store.
- For a given training resource, novices are able to explore a broader range of issues and to compose more coherent models of greater complexity.
- Those with prior experience have benefited most from the desktop. With access to tutorials and a limited interaction with an expert, such users have been observed to make reasonable progress in a matter of days.
- For those who require occasional access to simulation models, the clarity and consistency of the interface and the in-built documentation and attribution has eased the task of "coming back to speed".
- The tool-led are only disuaded from chaos by training, preferably in the context of a project and under close supervision.
- The nominal complexity of models has increased by an order of magnitude (whether such complexity is justified is another matter). More importantly, it has been possible to more effectively alter models to correspond with changes in design focus.
- Users' trust is fragile. Inconsistencies, uneven support for various simulation tasks and gaps in on-line help facilities can become critical failings from the users' perspective.
- An illusion of (relative) ease of use can mask an underlying complexity which users may not be equipped to deal with. Novices assume they have considerable skills— until they are called upon to exercise judgement based on the underlying physics.
- Novices continue to have difficulty associating what is displayed on-screen with underlying thermophysical processes.

Looking back over the case studies, the availability of project management facilities would have mitigated many of the difficulties encountered. Users would have made fewer errors, tedium would have been reduced and less effort would have been required to manage projects, but it is uncertain that software interventions would have altered the approaches to simulation that were adopted at the time. However, those who consider their approaches to simulation and document their work have fewer barriers to their support of design projects.

References

Bailey R.W, *Human Performance Engineering: Using Human Factors/ Ergonomics to Achieve Computer Usability (Second Edition), Prentice-Hall, Englewood Cliffs, New Jersey, 1989.*

Beranek D., Lawrie L., "Promising (and not so Promising) Developments in Energy Analysis Software" *Proceedings IBPSA BS '89* Vancouver, BC. pp5-10, 1989.

CIE, "Discomfort Glare in the Interior Working Environment", *Publication CIE No 55 (TC-3.4)*, International Commission on Illumination, Paris France, 1983.

Clarke JA, personal communication 1992.

Clarke JA, Hand JW, Hensen JLM, Johnsen K, Wittchen K, Madsen C and Compagnon R. "Integrated Performance Appraisal of Daylight-Europe Case Study Buildings" *Proc. 4th European Conference on Solar Energy in Architecture and Urban Planning*, Berlin. H.S. Stephens and Associates, Bedford, U.K., pp370-373, 1996.

Clarke JA, Hand JW, Janak M. "Results from the Integrated Performance Appraisal of Daylight-Europe Case Study Buildings", *Proc. Right Light 4*, Copenhagen, Denmark, 19-21 November 1997.

Clarke JA, Hand JW, Janak M. "Integrated Performance Appraisal of Daylight Buildings", *Prod. Daylighting '98, An International Conference on Daylighting Technologies for Energy Efficiency in Buildings*, Ottawa, Ontario. pp71-78, May 11-13 1998.

Clarke JA, Janak M, Ruyssevelt P. "Assessing the Overall Performance of Advanced Glazing Systems", to be published in *Solar Energy,* Elsevier Science, London 1998.

Clarke J, "A Performance Summary of Energy Conscious Building Technologies", accepted for publication in *Proc. EPIC '98*, Lyon, France, 19-21 November, 1998.

Compagnon R. Personal communication, EPFL, Lausanne, Switzerland, April 1997.

Crawley D., Personal communication 1997.

Degelman LO. "Daylighting Design Tools - Do We Have the Right Stuff", *Prod. Daylighting '98, An International Conference on Daylighting Technologies for Energy Efficiency in Buildings*, Ottawa, Ontario. pp11-17, May 11-13 1998.

Donn M.R., "A Survey of Users of Thermal Simulation Programs" Proc. BS '97, Prague, September 8-10 1997, Vol III pp65-72, 1997.

Facet Software, *General Facet Manual* IES Limited, Glasgow, June 1995.

Fitzgerald E., Lewis J.O. (Ed.) *European Solar Architecture: Proceedings of a Solar House Contractors' Meeting, Barcelona 1995*, Energy Research Group, University College Dublin, 1996.

Hensen J.L.M. "Introducing Buiilding Energy Simulation Classes on the Web" ASHRAE Transactions. *Vol 104 Pt. 1, 1998.*

Howrie J. "Building Modelling: An Architect's View" BEPAC Newsletter, Building Environmental Performance Analysis Club, No. 12, p8, Spring 1995.

Janak M. "The Run Time Coupling of Global Illumination and Building Energy Simulations", *Proc. Daylighting '98, An International Conference on Daylighting Technologies for Energy Efficiency in Buildings*, Ottawa, Ontario. pp113-120, May 11-13 1998.

Johnsen K., Grau K, *TSBI3 Computer Program for Thermal Simulation in Buildings User's Guide (Version B08)*, SBI, Danish Building Research Institute, 1994.

LESO-PB|EPFL, *Glazing Design Support Tool*, EPFL, Lausanne, Switzerland, 1998.

MacDonald I. "Uncertainty Representations in Simulation", *ESRU Publication 98-9*, University of Strathclyde, Glasgow, 1998.

Mumaw G. Personal communication, Glasgow 1997.

Pickering J. "What Illich wanted", *Active Learning*, No. 2 pp9, 1995.

Sangster A. "World Wide Web - What can it do for Education?", *Active Learning*, No. 2, pp3, 1995.

Shneiderman B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Second Edition, Addison-Wesley Publishing Company, Wokingham, England, 1993.

Sonderegger R. "Building Performance Simulation in a Commercial Software Environment", *Proceedings BS '89*, IBPSA, Vancouver, BC, pp163-168, June 23-24 1989.

Tufte E.R. *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire, Connecticut. 1997.

Ward Larsen G., Personal communication 1996.

Ward Larsen G., Shakespeare R., *Rendering with Radiance: the Art and Science of Lighting Visualisation*, Morgan Kaufmann, San Francisco, California. 1998.

Zhao A., Cook J., Higgen N. "Online Learning for Design Students", *Association for Learning Technology Journal*, Vol 4, No. 1 pp69, 1996.

Chapter 6

CASE STUDY OF PROJECT MANAGER IN USE

6 Case study of Project Manager in use

> *If modellers don't start each simulation with an idea of the reasonable range of outcomes, they will be at a disadvantage in routing out careless errors and misunderstandings about how the simulation works. On the other hand, a modeller who resorts to unrealistic inputs to generate what appear to be the "right" outputs may be working from erroneous preconceptions. It is important to know when to stop arguing with the model and to start listening [Kaplan 1992].*

This chapter provides a case study on the use of the Project Manager in a project of realistic complexity and focuses on:

- simulation as a central repository of information in the design process;
- simulation in support of integrated thermal and daylighting assessments;
- the resource requirements for various stages of a simulation-based project;
- the degree to which integrated performance assessments can enhance design decision support;
- how information from integrated performance assessments can be delivered to the design professions;
- the implications these assessments have for the cooperative use of tools.

6.1 Description and scope

The project is a design study for a low energy exhibition centre, the Brundtland Centre in Toftlund, Denmark, shown in Figures 6.1 and 6.2, which exemplifies attempts by the design professions to meld passive and active solar design strategies. It is based on work carried out as part of the EC's Solar House and Daylight Europe programmes [Fitzgerald and Lewis 1996, Clarke et al. 1996]. The former project focused on the development of low energy, prestige exemplar designs; the latter project on the understanding of daylighting within buildings. The Brundtland Centre analysis involved integrated performance assessments, component optimisation, and support for the design of a building monitoring regime. The design brief of the Centre included several goals:

- a high standard of thermal comfort without the use of air-conditioning, and with limited heating costs;
- heating supplied by direct solar gain, radiators in the upper floor, heated floors in the exhibition and atrium spaces and a displacement ventilation system in the lecture theatre;

Figure 6.1 Brundtland Centre, principal features.

- a high standard of daylighting via borrowed light from the atrium, light directing blind systems and occupancy sensing controls of uplighting and blind systems;
- balancing of lighting power demands by photovoltaic modules incorporating a mix of active and blank (where self shading was anticipated) glass-encapsulated modules in the sawtooth atrium roof, with three banks of modules on the south-east facade to enhance morning power delivery.

Some aspects of the design involved combinations of novel techniques and technology not amenable to conventional assessment techniques:

- The office block and exhibition spaces face southeast with little or no external shading, and light-directing modules in the fenestration limit glare and overheating.
- The Centre incorporates photovoltaic modules on the south-facing sawtooth roof to diffuse lighting and suppress glare sources in the large roof aperture of the atrium.

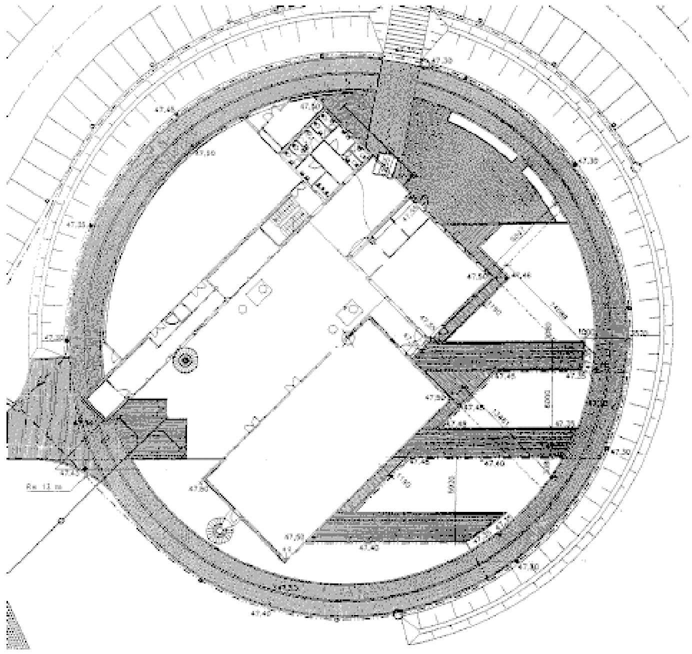


Figure 6.2 Brundtland Centre site.

- The Centre includes three active heating schemes in addition to direct solar gain, controlled ventilation between zones and the atrium as well as heat recovery units in the atrium and offices. In the summer, a night purge and a variable flow of fresh air into peripheral zones via the atrium is used to control overheating.
- The building shell is a mix of lightweight and massive elements, the latter used to moderate temperature swings and take advantage of direct solar gain.

6.2 The approach taken

The approach taken in this project was to proceed from the design goals and architectural and operational descriptions to define the metrics of the study, the rules for abstracting the design to support these metrics and the sequence of assessment tasks. The metrics of the project were thermal comfort in the exhibition and office areas, control of glare, lighting energy use, photovoltaic power production and overall energy demands. In addition, simulation support was to advise on the placement of sensors for a monitoring programme.

The rules of abstraction for this building were governed by the need to undertake detailed daylighting and glare assessments in monitored offices, indicate to the monitoring team the sensitivity of sensor placement, ascertain comfort levels in the exhibition spaces and determine overall energy use. The central role of the ventilation scheme implied that air flow required explicit treatment. It was anticipated that several models would be required, to isolate the contribution of particular facets of the design and to test design variants.

The intent was to develop a model which focused detail on a few selected rooms, used a moderate level of detail in adjacent areas and low resolution in the balance of the building. The resulting model, shown in Figure 6.3, focused on a typical south-east office, an atrium facing office, the atrium and PV modules, with other parts of the building represented at a level of detail sufficient to define thermal and visual boundary conditions.

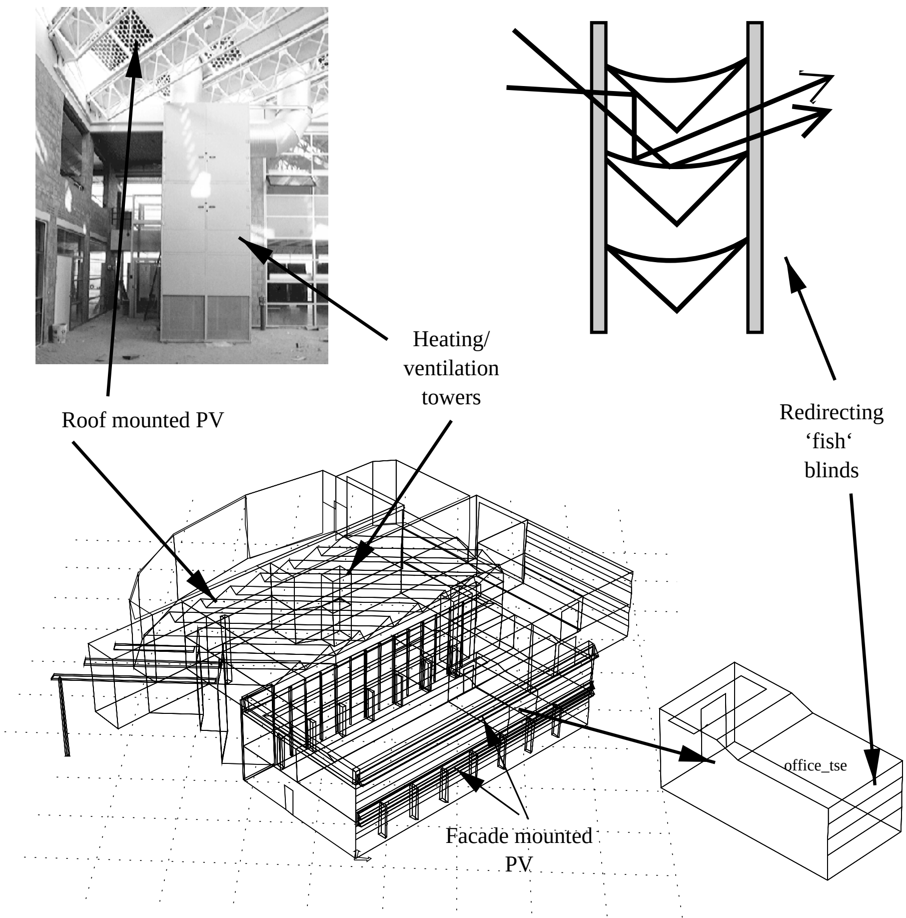


Figure 6.3 Brundtland Centre model.

6.3 Simulation tasks and results obtained

Several aspects of the model required critical planning. Temperatures and light levels in the offices and exhibition spaces were dependent on conditions within the atrium. The atrium was not well represented as a single space because temperature stratification was anticipated and the geometric complexity of the saw-tooth roof and bounding offices could not be ignored. The roof and facade mounted PV modules required an explicit representation which could be modified to support alternative assessment. Lastly, a mixture of forced and natural ventilation needed to be represented.

To test the sensitivity of project resource requirements to styles of user, two alternative approaches to geometric definitions were tested by two modellers. The first utilised CAD to input most of the building geometry with additional detail and attribution added within the Project Manager. The second approach used the Project Manager for all descriptive tasks.

After reaching a consensus on the broad composition of the model and its layout, the two modellers proceeded to plan their specific approaches to the descriptive tasks and to develop models with and without the use of CAD. The two models were reviewed and then testing and calibration proceeded on the non-CAD model. Syntax checking made use of printed reports, trial simulations and frequent checks of model details via the Project Manager. The approach taken to understand the building performance was to run one week assessments for each season and then review both aggregate and detailed performance predictions. Once the base case design was understood, reference models were prepared and integrated assessments undertaken.

The Brundtland Centre design team wished to guard against unintended consequences of design decisions and so it was important to assess a number of aspects of performance in the base case and each of the design variants. Integrated performance views (IPV), as described in Section 5.3, were used to achieve this.

Daylighting performance was also an issue in this case study. Facilities for the definition of thermal and visual models were described in Section 5.4. The Brundtland Centre assessments used these facilities to support visual assessments (e.g. visual comfort indices and glare assessments, daylight factor distributions as shown in Figure 5.44) as well as lighting control based on internal illumination levels. To support visual comfort analysis, it was necessary to include details of the desk, the target sheet of paper, parabolic lighting fixtures and round columns.

6.4 Observations

The observations from this case study relate to the resource required to undertake the work, the efficacy of the two approaches to model composition and the Project Manager's overall support of the simulation process.

Initial planning and data gathering required 12 hours. The CAD approach modeller spent two hours in detailed planning and sketching, and the balance of the day for CAD input and model translation. This collection of surfaces was then converted into a thermal model (e.g. adding and reconciling details, establishing topology and attributing the model). The fully attributed model (excluding flow and power networks) was completed at the close of work on the third day.

The second modeller spent five hours in detailed planning, model sketching and preliminary documentation. A fully attributed model (excluding the flow and power networks) was available in the early afternoon of the third day. It is instructive to review the differences between these approaches.

The CAD layout proceeded quickly from rough sketches and the original set of digitized points. Decisions on detailed geometry were made as the CAD model was generated. In the event, several zones were overly complex and needed to be subdivided. Adding the sawtooth roof required further adjustments to geometry. Topology checking was delayed until the geometry was complete and this indicated additional inconsistencies to be resolved (e.g. missing or inverted surfaces). Attribution proved tedious because the default surface, zone and component names provided little contextual information.

The second approach proceeded at a slower pace with the first morning given over to detailed planning of the form and composition of each zone and the production of detailed sketches with critical dimensioning added. The descriptive process proceeded at a steady pace with only minor reappraisals of approach and corrections. Surfaces were named as they were created and followed a standard naming and edge ordering convention. The topology-checking module was invoked at intervals (typically after three or four zones had been added) to identify inconsistencies and limit error propagation. Construction attribution was done last and made use of an "attribute many" option in the Project Manager.

Key observations from the model creation phase were that individual zones and surfaces were visited repeatedly (e.g. to update and check attributes) and the ad hoc nature of model instantiation requires continuous traverses of the interface. Such tasks make considerable demands on short term memory. In the CAD approach the time taken to traverse the interface, confirm the context of entities and check for errors extended as the model detail increased. The second modeller made a number of tactical decisions in the design of the model and the ordering of tasks so that the nature of the "building blocks" was clear, so traverse time was less of an issue and fewer errors were made. It is altogether possible that a more considered approach to the use of the CAD tool would have limited the number of corrective actions required within the Project Manager. Again, this is an issue for training and perhaps checklists for the practitioner as to critical "early investments".

Testing and calibration appear to require two different views of the model, indeed two different mindsets. The first is the essentially mechanistic checking for errors, model consistency and adherence to the design brief. Within ESRU, such tasks are carried out by both the modeller and by a third party. Here, the additional clarity in reporting the model composition and facilities for on-line access to the model have allowed inconsistencies and errors to be more quickly identified.

One intent of the Project Manager was to support incremental refinements to models in response to initial simulations. For example, the lecture hall (curved room at the upper left of Figure 6.3) was modelled as a lower occupied space with a ceiling zone and the plant distribution and service areas were treated as separate spaces. The simulations indicated that the lecture hall could be treated as a single zone and the service rooms combined with little change in overall predictions. The combination of editing facilities and contiguity checking allowed this reduction in complexity to be quickly implemented.

6.5 Conclusions

This case study shows that those who approach simulation methodically and with due attention to the metrics of a project can use simulation to deliver useful insights and add value to the design process. Appendix C provides a synopsis of seventeen other simulation-based projects which the author has undertaken and which are indicative of the cost-effective deployment of simulation within the time-constrained context of design practice. With the exception of two research projects, each involved open tendering and the client interactions and reporting conventions associated with professional practice. All projects were completed within the agreed time and resource budgets.

The Project Manager has contributed to this in several ways—it changes the distribution of resources within simulation-based projects, it supports a better understanding of models and the thermophysical processes which they represent and it removes many of the penalties associated with the iterative use of simulation in the design process.

The redistribution of simulation project resources is the result of decisions to provide balanced support of all aspects of simulation work rather than focus on conventional ease-of-use issues or practitioners fixation on geometry. The variety of projects in which simulation was successfully deployed confirms that the approach taken is broadly applicable. In the context of the author’s experience with ESP-r (circa 1990), DEROB and BLAST (circa 1987) and interviews with the users and developers of other simulation tools, it is concluded that:

- the combination of attribution, logical naming schemes, descriptive text and images improves support for quality assurance and the checking of geometry, composition or operational regimes;
- this feedback combination ensures clarity in presentation, enhances understanding of the model and eases tool navigation;
- the consistency manager removes much of the burden associated with incremental changes to simulation models;
- the use of a central desktop metaphor allows the user to focus on tasks rather than tool syntax and allows novices to explore simulation functionality;
- the use of the project configuration as the nominal unit of exchange supports a virtual design support environment which is insensitive to the physical location of staff and computational resources;
- design projects of considerable complexity can be attempted with limited computational, staff and time resources.

These changes in simulation project resources offer the practitioner more time for "living with the model long enough to understand it". In addition, the case study indicates that the Project Manager supports the iterative refinement of models and the detailed views of performance needed to arrive at a better understanding of the "energy signatures" which exist within the performance data, but which tend to remain hidden from many practitioners.

This case study also demonstrated the efficacy of using formal definitions of project metrics to support integrated assessments. In making explicit the metrics of the project and the focus of assessments, the definition of an IPV within the Project Manager provides critical support to the planning of simulation studies. The subsequent use of these directives to drive the computational process and extraction of performance data is a powerful mechanism for ensuring the delivery of useful information to the design process. The encapsulation of sequences of tasks as procedural logic has a number of advantages over the use of ad hoc scripts and is a way forward. However, such approaches only go part of the way towards supporting a truly integrated assessment environment which is the subject of the next chapter.

References

Clarke J., Hand J., Hensen JLM., Johnsen K., Wittchen K,. Madsen C. and Compagnon R. "Integrated performance appraisal of Daylight-Europe case study buildings", *Proc. 4th European Conference on Solar Energy in Architecture and Urban Planning*, Berlin, H.S. Stephens and Associates, Bedford, U.K., pp370-373, 1996.

Fitzgerald E., Lewis J.O. (Ed.) *European Solar Architecture: Proceedings of a Solar House Contractors' Meeting, Barcelona 1995*, Energy Research Group, University College Dublin, 1996.

Kaplan M. "Guidelines for Energy Simulation of Commercial Buildings", ACEEE Summer Study 1992. Vol. 1. pp137-147, 1992.

7 Enabling knowledge based cooperative design

> *The implied promise [of computer support for energy conscious design] is that energy analysis—through future computer-aided building design systems—could become an integral part of the design process [Clarke 1987].*

With the advent of project management facilities, simulation-based design decision support is increasingly viable within the constraints of research and consulting projects. In terms of placing simulation in a context where it can work cooperatively with other computational actors the Project Manager is a step forward. Clarke et al. [1995] have argued that it is necessary to move beyond the tool-box approach (shown on the left of Figure 7.1), where the designer is expected to recognise a particular task, locate a suitable program, run it and interpret its output in order to inform the design process. In such a regime, it falls to the designer to know enough about the tool to use it appropriately. An alternative approach, where tools work cooperatively and are able to access the data describing the design, and give performance feedback in terms meaningful to the designer, is shown on the right.

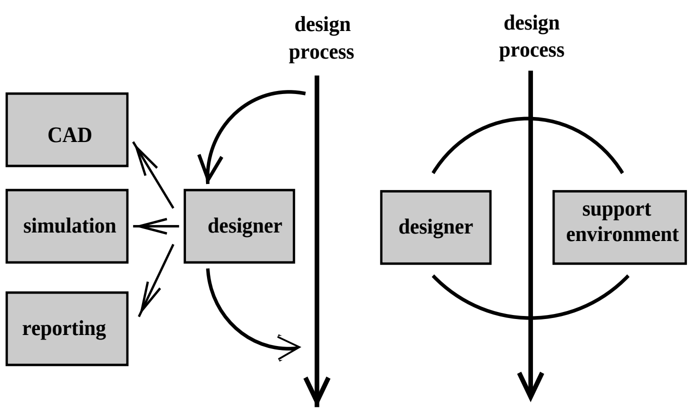


Figure 7.1 Tool-box and integrated approaches to design.

This chapter explores how the Project Manager might be extended towards an integrated environment where the control of the simulation process is guided by formal process models and explicit performance assessment methods. These extensions guard against the lapses in attention, extemporaneous approaches to simulation tasks which jeopardize project deliverables, and detect sequences of user actions which might have tactical or strategic implications.

The approach taken is to:

- introduce a knowledge-based agent within the simulation environment which observes the progress of the work and the current state of the model and tests them against criteria such as a project work plan or model of the design process;

- introduce a second knowledge-based agent which acts on observations of the current state of the work in order to inform the team when critical decision points are reached;
- formalise the conflation of information provided by other tools, and ensure that entities in each design tool are equivalent and that critical information is not lost as data is passed from tool to tool;
- generate an audit trail of user actions (e.g. decisions taken, facilities used, data exchanged, assessments undertaken).

A number of criteria are associated with such a computational support environment, hereinafter referred to as an Integrated Building Design System (IBDS).

- A consistent product model of a building and its systems is essential so that disparate design tools can obtain their models and return performance predictions.
- Design tools must take instructions from both the user and the IBDS.
- The interaction mechanism used must take a readily understandable form (e.g. functionality which is not applicable is removed from the desktop and available options are clearly delineated).
- Design tools within the IBDS must produce an audit trail so that the transactions (e.g. user to design tool, design tool to design tool) can be recorded and appropriate courses of action (strategic and tactical) inferred.

The work is founded on the Commission of the European Communities' COMBINE II project [Augenbroe 1992]. Details of the research are found in [Clarke et al. 1995] and Appendix E includes a description of this project and its deliverables to the building design professions. In order to clarify specific issues arising from the author's contribution, an outline of this earlier work is given in Section 7.1. This is followed by IBDS design issues in Section 7.2 and the formal data decomposition process in Section 7.3. Two facets of the project which have particular relevance to the current work—process modelling and knowledge based control—are described in Sections 7.4 and 7.5.

7.1 Review of the COMBINE project

The COMBINE[1] Project objectives were to develop an integrated data model (IDM) to hold a unified description of a design, to establish a mechanism to support the exchange of data within a representative set of design tools, to create an IBDS to coordinate access to a set of design tool functions (DTF) (e.g. altering building fenestration, undertaking overheating assessments). ESRU and Rutherford Appleton Laboratory worked together to:

- derive a formal description (aspect model) of the ESP-r data model;

---

[1] Computer Models for the Building Industry in Europe

- enable facilities and schemas to exchange the data model with the IDM;
- use rapid prototyping to create an IBDS desktop and the underlying communications and process monitoring conventions;
- explore the introduction of design process knowledge and extend the IBDS with knowledge based control of design sessions.

The aim of the IBDS is to support realistic design sessions and then configure and observe different design approaches under knowledge based control. It is built on the framework of the Intelligent Front End (IFe) [Clarke and Mac Randal 1991] to allow rapid prototyping of different interaction forms and access to knowledge based control.

The IBDS incorporates several real design tools (AutoCAD Release 12 [Autodesk 1989], MicroStation [Conforti 1994], the ESP-r Project Manager and Simulator, regulations compliance via BRC [Rode 1993], TSBI3 [Johnsen and Grau 1994] and Daylight/ Visual Impact via Radiance [Ward 1993]). The desktop is shown in Figure 7.2. Beginning clockwise from the upper left the desktop provides selections for session details, design mode, user type (non-expert architect), feedback, and on the desktop are icons for tools which are currently relevant. Intra-tool chatter in the background window would normally be hidden but is included in the figure to show the messages passing through the communications centre.

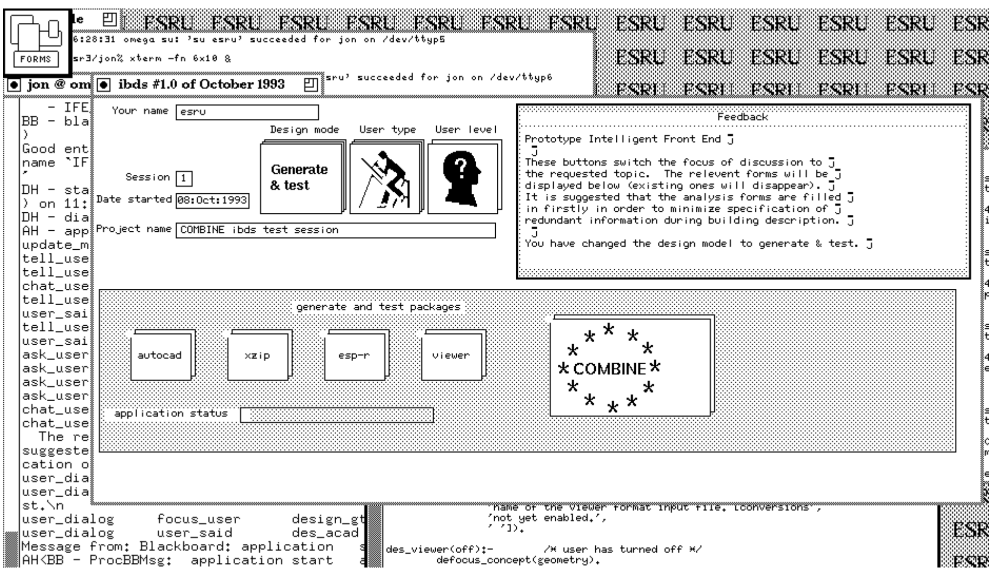


Figure 7.2 IBDS desktop.

In a conventional palette of tools, the user would be required to select an applicable tool, determine what information was required and if it was current, translate or filter it into the appropriate format and, after the design session, inform colleagues of the changes and distribute the revised model. In contrast, such tasks are accomplished by the IBDS.

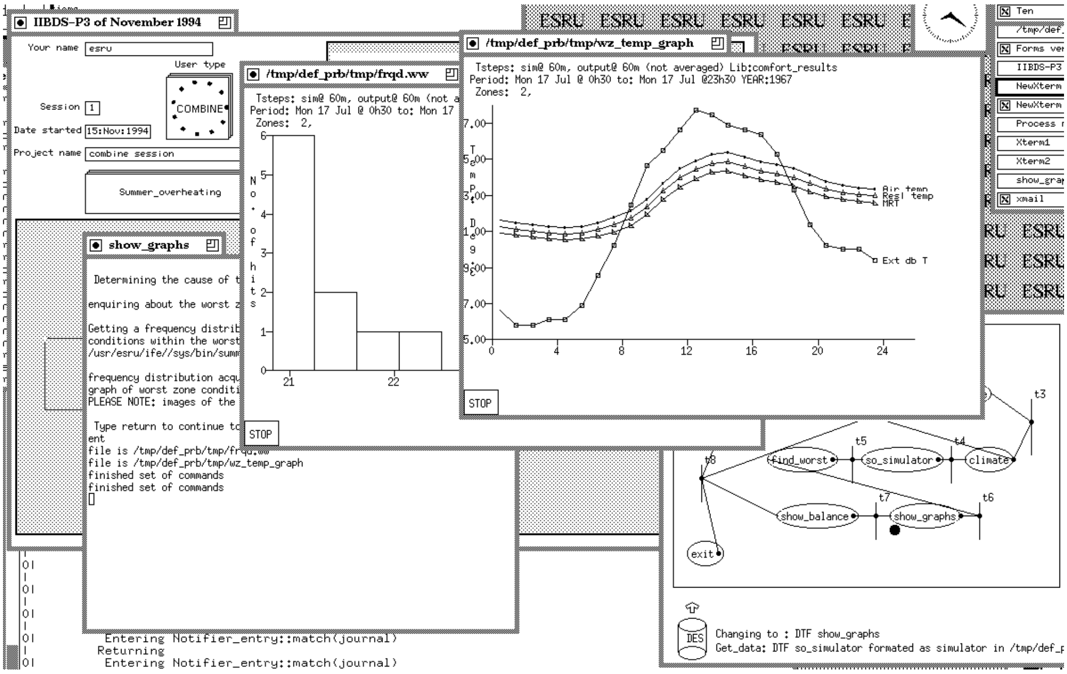


Figure 7.3 IBDS desktop during a design session focused on summer overheating.

Of particular interest in the current work is the inclusion of knowledge based control. Figure 7.3 shows a design session focused on summer overheating risk. Here, in response to a predefined operational regime and performance assessment method (explained in Section 7.3), the desktop has invoked a sequence of interaction, analysis and reporting functions. In the figure the IBDS is determining what constitutes the worst spaces in terms of overheating based on the highest resultant temperature in an occupied space. Overheating has been detected and a frequency binning of temperatures and a graph of temperatures in the worst zone are presented along with the causal factors in the worst zone. The user can then either exit the design session or change the form or composition of the model. This process is reported by the process monitor (lower right of the figure) on the manager's workstation.

7.2 The design of the IBDS

The design of the IBDS is predicated on the separation of the user interface from the suite of design tools by a Blackboard-based communications centre and knowledge based core. The passing of messages within the system is the mechanism which enables autonomous actors to work together to accomplish useful tasks.

The internal structure of the IBDS is shown in Figure 7.4 and includes:

- a Blackboard-based communication centre with message passing between the user interface, knowledge bases and design tool functions (DTF) used to control tool access and record an audit trail (via the message monitor);

- a User Knowledge Handler (UKH) and Application Knowledge Handler (AKH) to control communication with the designer, access to the data model and design tools;
- a Design Tool Function Knowledge Handler (DTF KH) to map functional requests to the relevant DTF;
- an Application Handler (AH) to control the design tools, pass them their data and receive their returns;
- a Data Handler (DataH) to coordinate the exchange of data with the integrated data model (IDM);
- a Process Monitor to observe the exchanges within the system, invoke design decision dialogues with the user and inform the manager of the project of its current status;
- a formal description of the process models which are available for the design session (Petri networks).

This design allows the configuration of the IBDS to be determined by a list of design tools and a formal design process description, a knowledge base which controls access to the integrated data model and a process monitor. This is explained in Sections 7.4 and 7.5.

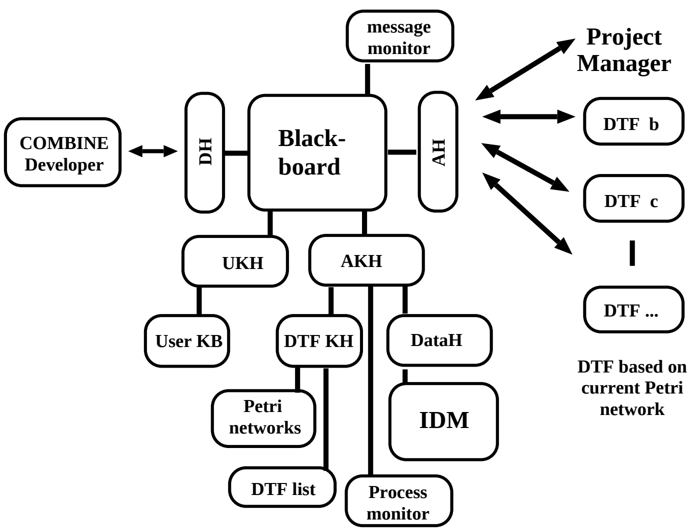
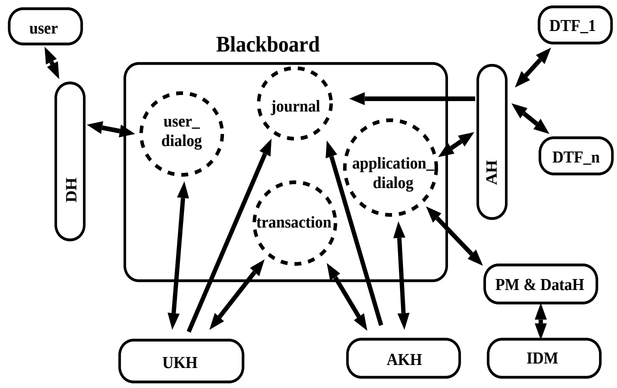
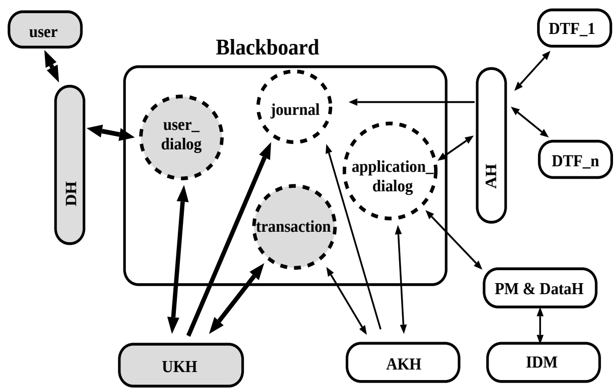
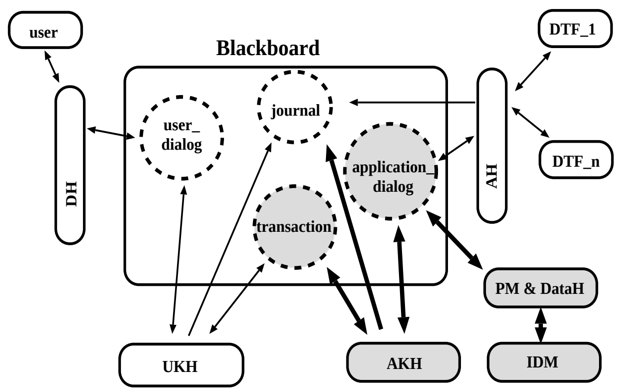
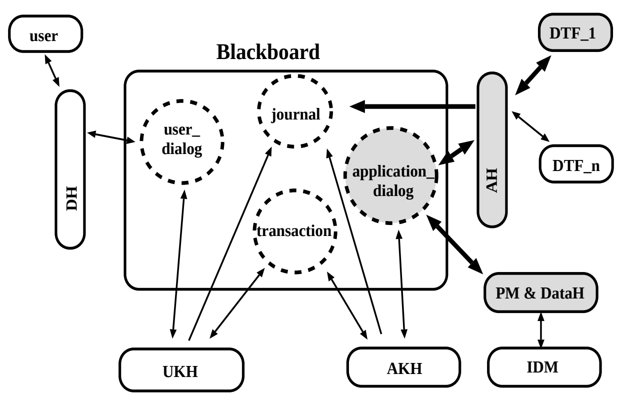


Figure 7.4 IBDS architecture.

One way to explain the relationships within the IBDS is to follow interactions during one phase of a design session. This is represented as a series of system snap-shots in Figure 7.5. The arrows show the potential flows of information: a single arrow indicates a notification while double arrows indicate sending and listening. Shaded boxes indicate active portions of the IBDS at each step. The "user_dialog" area of the Blackboard is reserved for user interaction transactions, while the "application_dialog" area is reserved for transactions related to the design tool functions. The "journal" area receives messages from the various knowledge handlers and organises these for subsequent analysis and process control.

<table>
<tr><th>IBDS</th><th>Description</th></tr>
<tr><td>
</td><td>This is the initial state of the IBDS before any user interactions.
The following abbreviations are used:
Dialogue Handler (DH), User Knowledge Handler (UKH), Application Knowledge Handler (AKH), Data Handler (DataH), Application Handler (AH).</td></tr>
<tr><td>
</td><td>1) A message passes to the DH indicating the requested interaction
2) the DH passes the message to the "user_dialog" area
3) and the UKH tells the Blackboard to "start DT" (where DT is the design tool function name).</td></tr>
<tr><td>
</td><td>The UKH issues "application_dialog start DT" to the transaction area.
1) The AKH finds the application and posts the message "new_application DTx" to the "application_dialog" area.
2) The DataH then queries the IDM - "get_data_for DTx" - and the IDM returns the appropriate data for DTx as "data_for DTx file".</td></tr>
<tr><td>
</td><td>The IDM issues "data_for DTx file":
1) The DataH posts "new_application DTx (parameters)" to the application dialog area.
2) The AH starts the application and establishes a pipe to receive the performance return(s).
3) When complete the AH records this fact and sends "closed DTx revised_data_file" to the application_dialog area.</td></tr>
</table>

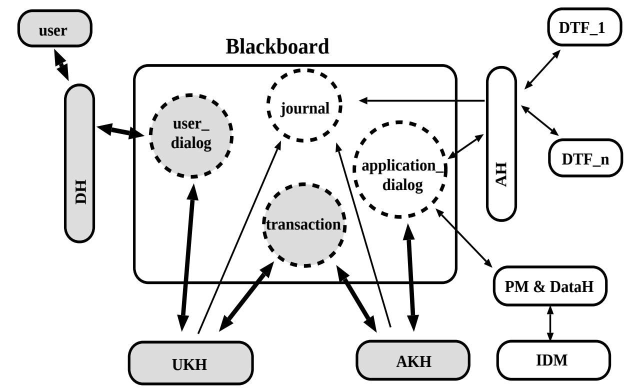
IBDS	Description
	The AKH posts "closed DTx" to the transaction area, which is received by the UKH for transmission back to the user interface.

Figure 7.5 Snapshots of the IBDS.

Upon invocation of the desktop the user is presented with a list of design sessions. The selection of a session causes the knowledge handlers to load the relevant process model and thus the root definition of what is possible within the design session and which "buttons" to include in the interface. After the user selects a design function, the knowledge base a) ensures that the data required for the task is available, b) starts the appropriate design tool at the appropriate point and then c) hands control to the user; it then monitors what the design tool is doing and finally ensures that the results of the tool are captured and propagated. While the IDM handles the data, the application knowledge handler retains responsibility for driving the process, propagating information to other knowledge bases and keeping track of the design status and history.

The message passing within the IBDS forms a detailed audit trail of tagged messages which form a potentially rich source for those who would seek to understand how tools are used to support the design process. This is covered in detail in Chapter 8. Further information on the underlying communication mechanisms can be found in Appendix E. The next section reviews the formal decomposition of the ESP-r data model.

7.3 Data model decomposition

As pointed out in Section 5.1.4, translations among different data models are a considerable challenge. The more defined the syntax, and clear the semantics of the source and target models, the higher the probability that a translation function can be written. The author has implemented a number of filters within ESP-r (see Table 2.3). However, as the number of tools increases, the problem quickly becomes intractable. The IBDS explored new forms of interactions within and between design tools in which the data exchange mechanism understood both the syntactic differences between tools and the underlying semantics of the data models.

In order to derive a unified superset model and communications protocol, a clear understanding was required of the semantics and relationships within each data model. In a data model as extensive as ESP-r and dispersed over some 245,000 lines of code, it is difficult to see such patterns and so a formal decomposition was required. The approach taken within COMBINE was to use the symbolism of ATLIAM (an extension of the Nijssen Information Analysis Method (NIAM) [Spiby 1991] to identify the semantics of entities and the relationships between them. The formal decompositions of all design tools were then conflated within the IDM and used to define the protocols of data exchange.

With regards to ESP-r, 54 ATLIAM diagrams were produced, of which six relate to performance (output schema), the rest to the input requirements (input schema). One input schema relating to the overall problem composition is shown in Figure 7.6. This diagram uses the symbols shown in Figure 7.7. Entities which are on more than one diagram have the CLONE symbol of a square box. The box between entities is equivalent to Condensation_risk HAS Condensation_details. The arrow between entities indicates that Condensation_risk is a subtype of Performance_assessment_requirements.

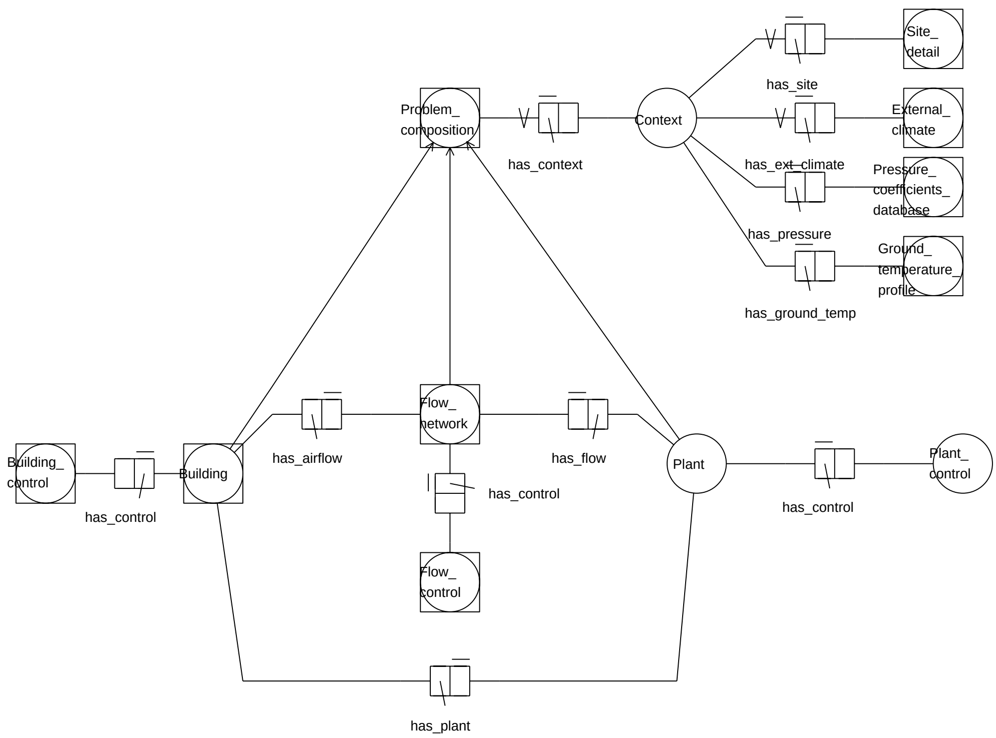


Figure 7.6 ESP-r Problem_composition ATLIAM diagram.

Figure 7.6 translates to: the Problem_composition is composed of a Context (mandatory) and, jointly or severally, Building, Flow_networks and Plant. A Building may have a Building_control, Flow_networks may have Flow_control and Plant may have Plant_control. The model Context must include Site_detail and an External_climate. There might be a Pressure_coefficients_database or a Ground_temperature_profile associated with the Context.

Such decompositions clarify the information required to support specific design questions. For example, a "performance assessment" is a set of data supporting a performance report which has been

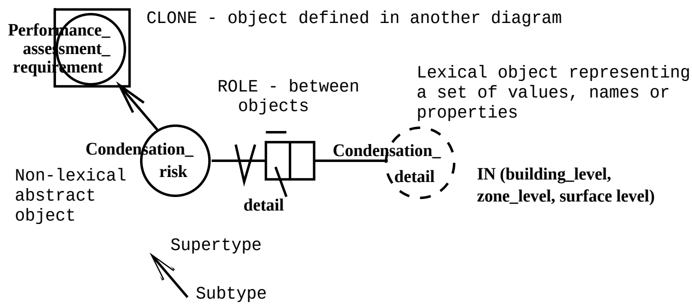


Figure 7.7 ATLIAM symbolism.

generated against a given design intent. This concept was later developed into the formalism of an Integrated Performance View. The process also uncovered inconsistencies within the ESP-r data model regarding the treatment of scheduling for controls, occupancy and flow. It also suggested where the data model can further evolve.

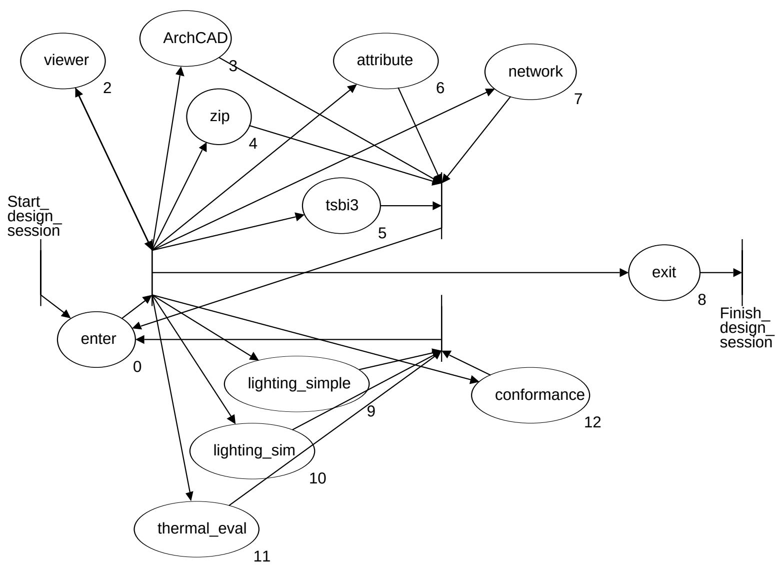
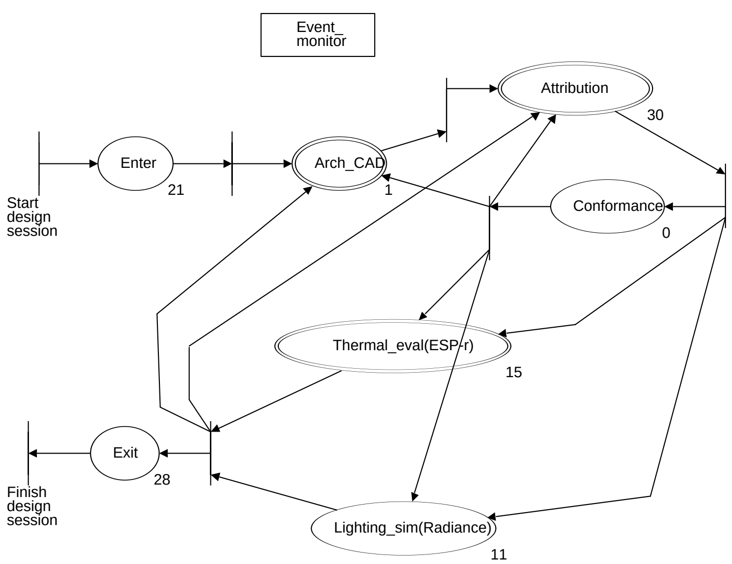
Further information on the decomposition of the ESP-r data model is given in Appendix B. A fully documented set of diagrams (updated for subsequent changes in the ESP-r data model) can be found in [Hand and Strachan 1998]. Sections 7.4 and 7.5 relate to the specific contributions of the author to the use of process modelling and the introduction of knowledge based control of design sessions.

7.4 Process modelling

With the Project Manager, control of the simulation process is vested in the practitioner and the (perhaps fortuitous) adherence to guidelines established in the project planning stage. A manager who finds a particular operational regime effective and knows which staff are suited to particular tasks might wish to configure the decision support environment to use some of the available design tools (and exclude others), enforce syntax checks at specific points during the project and make explicit a performance assessment method. The IBDS provides a mechanism for this and allows the manager to observe the current state of the model and the process of design support as it evolves. The intent of this is to give as much autonomy as possible to the personnel actually doing the work, delegating responsibility while maintaining overall management control.

Because this is similar to many of the issues arising in classic process control modelling it is useful to use the same formalisms. The act of capturing the elements of the desired process in a flowchart form, apart from requiring participants to be explicit about what they do, means that the "flowchart" can be displayed, designed (not just evolved), reasoned about and communicated. The formal flowchart used in the work is a variant of a Petri-Net [Javor 1993]. Petri-Nets were used to define the sequencing of tools and decision points in design sessions. The formal underpinning of a Petri-Net makes it possible to check for inconsistencies.

Figure 7.8 shows the Petri-Net expressions of a number of design sessions used within the COMBINE project. The last one corresponds to the summer overheating scenario shown in Figure 7.5. The symbolism used is: vertical bars represent transition points, oval entities with double lines are design tools and those with single lines are design tool functions. The lines between ovals show potential choices and the arrows show progression.

Petri-Net	Description
	An unconstrained and unsequenced access to design tools. The user can approach any tool in any sequence (i.e. a typical professional office). The process model is used for tool access monitoring.
	An unconstrained, but sequenced design session (e.g. guidance is provided but details are left to staff). It progresses from an initial CAD definition, through attribution and conformance checking stages and then a choice of a thermal or lighting evaluation. Loop-back points are provided so that attributions can be changed or the CAD tool revisited.

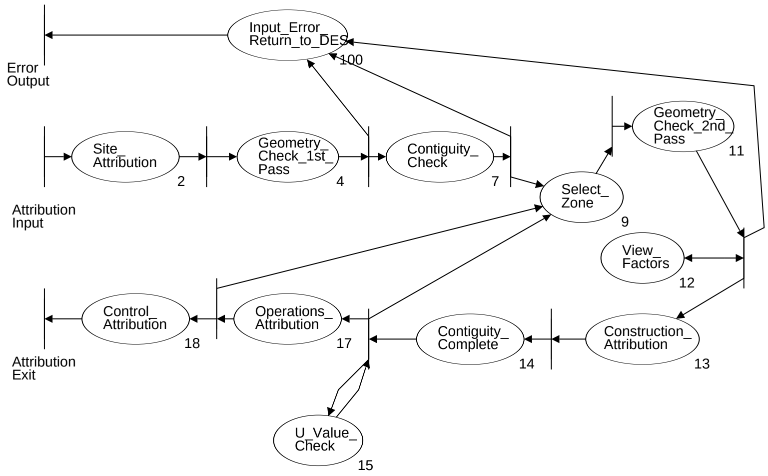
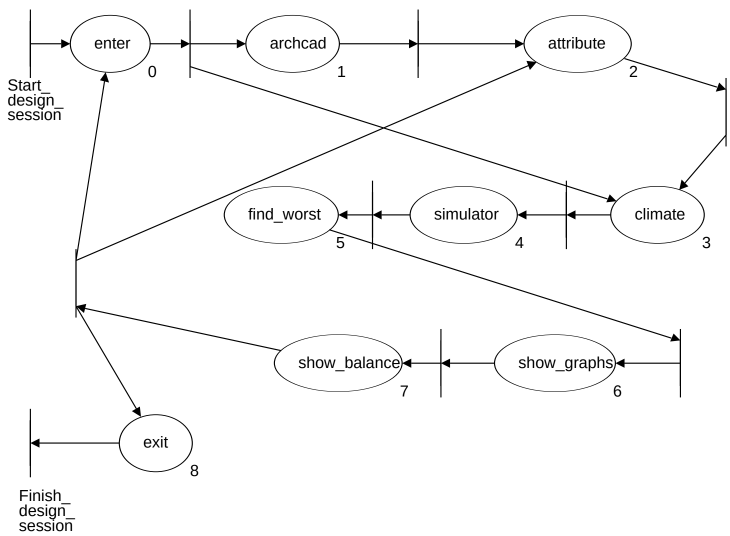
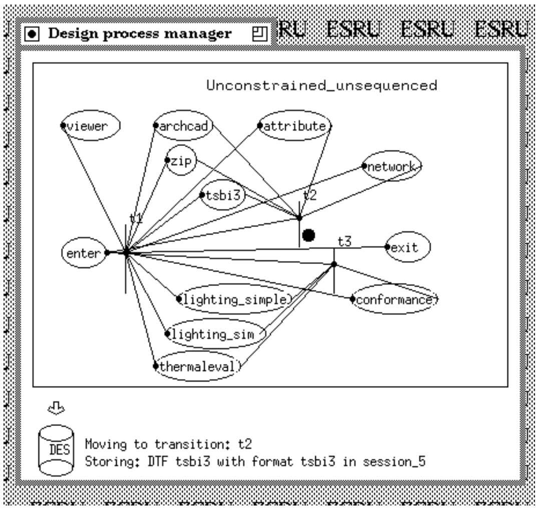
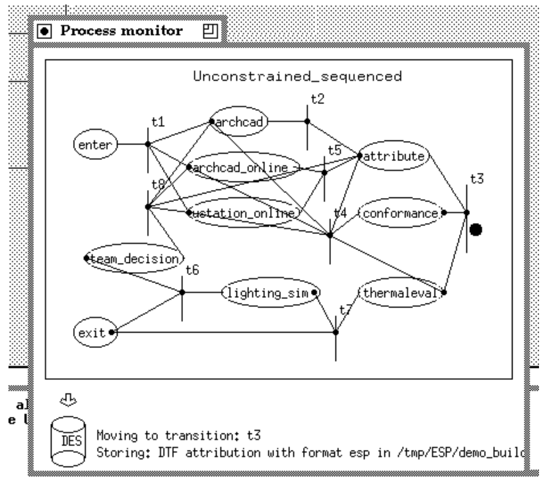
<table>
<tr><th>Petri-Net</th><th>Description</th></tr>
<tr><td>
</td><td>Expanding the "Attribute" phase (above), to demonstrate steps suitable for quality assurance. This model of the design process insists (for good reasons) that Site_Attribution be carried out early in the process. Such sequences can assist in the design of tools (e.g. to provide relevant feedback at different stages).</td></tr>
<tr><td>
</td><td>This represents a summer overheating session. After attribution a relevant climate and assessment period is selected, a simulation is commissioned and a "find_worst" search undertaken. This is followed by a presentation of supporting graphs and energy balances. Depending on the outcome of this, attributes of the model can be revised.</td></tr>
</table>

Figure 7.8 Design session Petri-Nets.

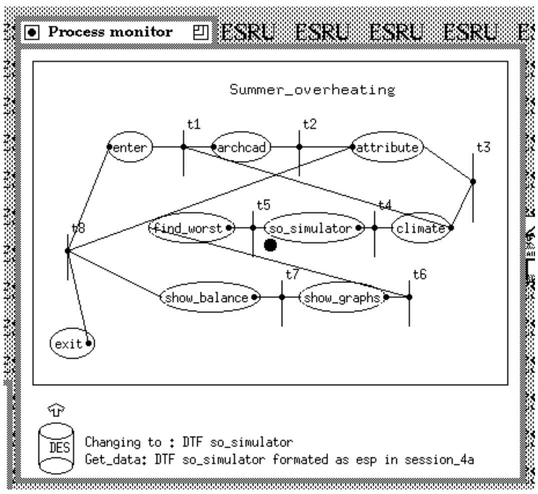
The unconstrained and unsequenced regime in Figure 7.9 represents a ‘shallow’ control scenario with few interactions. The second process imposes a sequence on tasks but possesses no knowledge about design purpose. The summer overheating process affords ‘deep’ control based on knowledge of a design purpose. The system, not the designer, controls the order of tool selection and the invocation of functions. A manager could use this to enhance quality assurance or to enforce checking of particular issues such as lighting performance. Self-directed teams could use such controls to signal for consensus meetings and a consultant could check progress against initial estimates.





The upper left process diagram matches the unconstrained and unsequenced Petri-Net. The "token" is after the tsbi3 tool and at the point where performance results have been stored. The manager is now in a position to ask for a report on the latest performance assessment.

The upper right process diagram matches the unconstrained and sequenced Petri-Net. The "token" is after the attribution phase and the current model has been returned to the IDM. The manager could now browse the model and extract the current area of glazing.



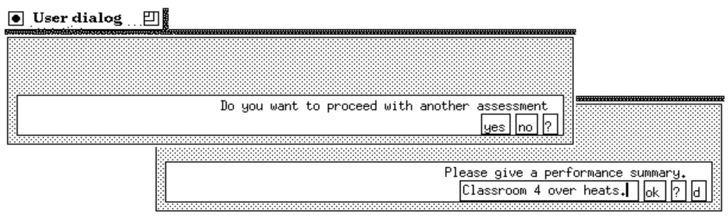
The lower left process diagram matches the summer overheating Petri-Net. The "token" is after the simulation has been run and prior to the "find_worst" task (i.e. wait a few minutes before asking about the frequency of discomfort).

Figure 7.9 Design session process displays.

To allow the manager to observe the design process the author created a visual Process Monitor and developed a process display description language. The Process Monitor displays the current Petri-Network with the addition of a "token" which traverses the nodes and an IDM "extract/ store" icon. The scenarios in Figure 7.8 are shown in Process Monitor form in Figure 7.9. In order to extract a model from the IDM the process monitor checks the Blackboard for messages of the form of `get_data_for(design_function_name, format);` or `store(design_function, tool_name, format, file_name);` to update the IDM. The Process Monitor responds to `token_move({place_name|transition_name});` by moving the Petri-Net token to the appropriate position. In the absence of an IDM the process monitor invokes and coordinates the filters (see Table 2.3) used to generate an instance of the current model in the appropriate format for the tool to be invoked.

The transition points in the Petri-Nets are where decisions are taken and the model is passed between design tool functions. In most cases decisions are devolved to the user (supported by relevant displays of performance data). Decisions are supported by the dialog tool shown in Figure 7.10. It is passed the choices to be made and supporting text (see calling conventions in lower portion of Figure 7.10) and this allows it to be driven from the Application Handler, a script (wrapper around a design tool) or one of the design tools.

Because some design tools can not be modified to assist the IBDS in its tasks they are encapsulated in a "wrapper" which usually takes the form of a shell script which runs a simulation, checks if it is completed, asks for the name of the results file created, starts the results recovery tool and then asks for a synopsis of the analysis.




```
                         dialog -help
 Use: [-p <quoted prompt string>]  {-file|-s|-num|-ab|-ok}
      where -file <quoted string <= 72 char> -num <number>
        -s <quoted string> -ab <quoted choice a> <choice b>
        -ok yes & no selections provided
      [-d <quoted default response>
      [-h1 "help text"] [-h2 "help text"] [-h3 "help text"]
  -help :this help message.
# [example of] Ask user for summary of performance.
schema=‘dialog -p "Please give a performance summary." -file " "
-d "Classroom 4 overheats." -h1 "Give a brief statement on the
analysis results" -h2 "A one line summary is required."‘
[returned string] Classroom 4 over heats.
```

Figure 7.10 Dialogue tool and calling conventions.

7.5 Process knowledge

The choice of design tools presented to the user and the way they are sequenced and constrained is supported by the addition of design process knowledge (i.e. the Petri-Net for the current design session). Each node in the Petri-Net corresponds to a design tool function and is associated with knowledge of what could or should be done at each point. To associate knowledge with the nodes, the Petri-Net is encoded as a series of "transition" facts in the Prolog language in the format: `transition (df1, df2, trn);` where "df1" and "df2" are design tool functions such that the completion of df1 enables the execution of df2 by way of transition "trn". The Prolog facts related to the unconstrained sequenced design session Petri-Net (upper right of Figure 7.9) are listed in Figure 7.11. Each design tool function has an equivalent "place" in the diagram.

```
%% project window for unconstrained/ sequential design session.
 transition(enter, archcad, t1).
 transition(enter, archcad_online, t1).
 transition(enter, ustation_online, t1).
 transition(archcad, attribute, t2).
 transition(archcad_online, attribute, t5).
 transition(ustation_online, attribute, t5).
 transition(attribute, thermaleval, t3).
 transition(attribute, conformance, t3).
 transition(conformance, attribute, t4).
 transition(conformance, archcad, t4).
 transition(conformance, archcad_online, t4).
 transition(conformance, ustation_online, t4).
 transition(conformance, thermaleval, t4).
 transition(thermaleval, lighting_sim, t7).
 transition(thermaleval, exit, t6).
 transition(lighting_sim, team_decision, t6).
 transition(lighting_sim, exit, t6).
 transition(team_decision, attribute, t8).
 transition(team_decision, archcad, t8).
 transition(team_decision, archcad_online, t8).
 transition(team_decision, ustation_online, t8).
```

Figure 7.11 Petri-Net encoded as Prolog facts.

At the beginning of the design session the IBDS is configured by loading such lists of facts. It then offers the user design tool function "buttons". When selected, the appropriate transition is made and the target design tool function is invoked (by posting a request to the Transaction area on the Blackboard). If at a later point in the design session another operational regime becomes more appropriate, the IBDS can be reconfigured by loading a new Petri-Net (set of Prolog facts).

The "transition" facts in Figure 7.11 are processed by a Prolog predicate named "next", each time the token moves to a transition. As a token passes from a place a similar predicate "next_trn" is used to produce a list of possible transitions. Both predicates in Figure 7.12 are deceptively short.

```
%% return the list of possible next choices
next(_current, _nextlist):-
  bagof(_next, transition(_current, _next, _trn), _nextlist).
next(_current, _nextlist):-
  _nextlist = [].
%% return the next possible transition
next_trn(_current, _trnlist):-
  transition(_current, _next, _trnlist).
next_trn(_current, _trnlist):-
  _trnlist = [].
```

Figure 7.12 Next predicate.

In turn each of the design tool functions is specified via Prolog predicates such as those shown in Figure 7.13. The parameter (get_data_flag) tells the IBDS whether information is required from the IDM for a particular (design_function) which is carried out by the application (toolname) and requires data of (format) type files and to be run on the machine (domain_name).

```
%% get_data_flag, design_function, toolname, format, domain_name
tool(data_req, archcad, autocad, autocad, skye).
tool(data_req, archcad_online, autocad_online, autocad, skye).
tool(data_req, ustation_online, ustation_online, autocad, local).
tool(data_req, attribute, esp, esp, local).
tool(no_data_req, conformance, uv_chker, none, local).
  ....
```

Figure 7.13 Prolog predicates for design tool functions.

The code fragment in Figure 7.14 shows the knowledge based invocation of a design tool function. After the message is passed to the Blackboard to start the DesignFunction, the predicate records that the design function is live, to avoid conflict with other design functions, and finds out what options might be applicable when the design tool function is complete and then informs the user. Other knowledge bases intercept the start request and collect the appropriate data from the IDM, and start the application. The explicit and sequential nature of the design session process models ensures that concurrency (managing parallel threads which access or update the same data) is rarely an issue in the IBDS.

```
...
design_function(start, _DesignFunction, _Description):-
  thread ask_usr(_DesignFunction, on),       %% display as selected
  to_bb(transaction, start, _DesignFunction), %% message to start
  assert(runing(_DesignFunction, _Description)),  %% remember which started
  next_trn(_DesignFunction, _trnlist),        %% see what is next
  transition assert(shift_transition(_DesignFunction,
  _trnlist)),        %% remember next transition
  chat_usr([[ _Description, ' is now running.'],' ']).  %% tell user tool running
....
```

Figure 7.14 Design function (start) predicate.

After a design tool has been invoked, the user interacts directly with it. The tool then marks its progress by sending messages back to the Application area of the Blackboard. When the tool is finished other predicates manage the transfer of new information to the user, and, where appropriate, to the IDM.

Although design purpose is handled by the IBDS, design decisions (e.g. reconciling direct solar gain in terms of heating potential and glare discomfort) must be made by the designer, based on information provided by the IBDS. Depending on the methods used by the design team or imposed by regulatory bodies there may be specific assessment sequences available to support such decisions.

7.6 Observations

The degree to which the IBDS empowers the design process is largely a function of how well it lets practitioners do what they would otherwise have done by more laborious means and then adopt new working practices. Observations of the IBDS and comments from design professionals indicated that it was a radical departure from traditional uses of design support tools and had considerable implications for design practice.

When the IBDS was first demonstrated in 1994, the use of a process model to influence a design session was a radical concept. That the behaviour of a computational support environment can be reconfigured on the basis of a compact set of instructions continues to surprise observers. At the time of this writing no other simulation-based design decision support system implements the notion of deep process control.

Once the IBDS is installed, a knowledgable practitioner can easily alter a process model to reflect the inclusion of new design tool functions or decision points. In most cases where shallow control is needed, only a few moments are required to configure a suitable "wrapper" and an additional transition point dialogue. Although the introduction of new knowledge may require short additions to a number of knowledge bases and configuration files, the process is essentially a cut and paste exercise for those who are *au fait* with the system. The communication and control mechanisms support the explicit representation of a variety of performance assessment methods, usually without the need to alter the functionality of the design tool. This has implications for professional firms with well-established working methods and quality assurance procedures.

The benefits (from the process modelling point of view) of mapping separate design support functions to separate tools allows a number of bespoke tools to contribute to design decision support. For example, this does not preclude using an integrated simulator to provide the functionality of a U-value checker or a condensation risk tool or a code compliance module. What is important is the delivery of the specific information required at a point in the design process.

The maintenance of data integrity during design sessions was, for many observers, the core benefit. For practitioners a transition between two tools is not necessarily an unremarkable experience. To shift between four tools in as many minutes engendered considerable interest. It presents the possibility of selecting the most appropriate tool for a specific task rather than compromising with an inadequate tool.

The formal decomposition indicated a considerable degree of commonality within a set of performance assessment tools. This has implications for the development of simulation methods and skills acquisition which are discussed in Chapter 8.

These advances were accompanied by a number of difficulties which limited the efficacy of the IBDS. These ranges from the pragmatic (e.g. the lack of attribution in CAD tools) to the fundamental need for design teams to change the way they work:

- Even as a prototype it was possible to observe user actions (e.g. changing fenestration details) being distributed across a range of design tools.
- Transactions with the IDM proved one or two magnitudes slower than filter based data transactions and the associated computational infrastructure proved to be considerably demanding of disk space and licensing fees.

- Often there was no direct mapping function to extract frequently demanded data from the IDM. Formal methods do not always identify the practical demands of data exchange.
- Petri-Net based process models work with "flow", and can have only limited "context dependency", so choices of what to do next require user intervention (e.g. if acceptable send to the Planning Department, otherwise return to the Architect).
- The delivery of a conceptually simple process model definition relies on a dozen computational actors working cooperatively and as many entity and file naming conventions being followed within the system.
- The ESP-r desktop does not always "listen" to child processes and thus the implementation of some "deep control" directives proved difficult.
- Work-flow processes may fall apart if one component of the process unexpectedly fails (the key word here is "unexpectedly").

7.7 A comparison of approaches

In operation, there are many similarities between the functionality and relationships of computational entities within the IBDS and ESP-r’s Project Manager. For instance, the addition of a surface causes an update of the data model in each. Each provides a number of checks to validate entities and enforce model contiguity. The imperative for the Project Manager’s use of tight binding is similar to that of the IBDS desktop’s links to the integrated data model.

What is particularly intriguing is to compare the *frequency* and *granularity* of the links between displayed entities and the underlying data model. Recalling the actions of the consistency manager in Section 5.2.6, some user actions (e.g. adding a surface) required immediate reconciliation of the data model and others (e.g. recalculating shading patterns) were best delayed until the model was static. Tight binding implies a high frequency of transactions, and interactive exploration of design issues can easily result in iterative assessment cycles in the order of minutes.

The types and frequency of interactions within the IBDS have been compared with that of the cooperative use of thermal and lighting assessments described in Section 5.4. In the conventional scenario, the Radiance desktop was passed the current data model and this was converted into native Radiance format (without information loss and with full annotation and structuring of the data store). This was then manipulated along with Radiance executables (which were themselves unchanged) to act as an on-line design tool. In practice, the Radiance desktop acted to extend the design decision support functions available to the user.

With the IBDS the burden of communication and the prototypical nature of the on-line CAD interface constrained the functionality offered to the user. In contrast to the Project Manager’s combined use of text and graphic attributes, the on-line CAD tool made it difficult to ascertain the identity and composition of an entity.

The conventional approach of the Radiance desktop (e.g. procedural logic and filters) has the advantage of speed and functionality. With a different data exchange, implementation speed would probably not be an issue. It remains to be seen if CAD tools will evolve in ways which are more compatible with the needs of assessment tools.

The critical issue which separates conventional and knowledge-based approaches is process control. Although ESP-r modules are capable of generating an audit trail of user actions, decisions and facilities invoked, there is no agent which takes advantage of this information, no mechanism for influencing the practitioner's course of action and no mechanism for a manager to observe the progress of the work. This is not to say that forms of process control are not active within conventional tools, but the Project Manager could benefit from additional channels of control.

Another point of comparison is the mechanism for introducing new data model entities. In conventional settings this task is inevitably ad hoc in approach, costly to test and document and requires each transform or filter to be treated both in isolation and in combination. An IDM, by insisting on clearly defined formal approaches, is largely self-documenting and verifiable. Code interventions have the potential to be concentrated in a few points of exchange and amenable to automation. An optimistic view is that a superset model which benefits many users is also one to which many will be inclined to contribute over time.

7.8 Implications for simulation

The concept of a unified model and the coupled exchange of data and semantics does not address the question of whether the design process is well served by a single model within a palette of design tools. Issues of granularity, focus, and temporal evolution remain. Simulation is driven by rules of abstraction, attribution and topology which may not be relevant to other design support activities such as structural analysis. Simulationists tend to use relatively simple geometry in comparison with that used in architectural CAD even when attempting combined thermal and lighting assessments.

One goal of the current work was to address the lack of fit between general CAD applications and the information demands of assessment tools. The resource required to filter out spurious and topologically unrealistic constructs, apply attribution and establish topology has been a powerful argument against the cooperative use of conventional CAD tools and simulation. COMBINE addressed this by treating CAD as an on-line tool which constrained user actions to IDM entities to ensure that their semantics were known. What emerged was a different tool which enforced a different mode of interaction with the user. That the project demonstrated CAD sessions which generated information which could be directly applied in other tools is thus a considerable achievement. However, consideration must also be given to how this alters the nature of the CAD models design firms compose and if it compromises the traditional uses for such models. The optimistic view is that attributed and topologically correct models will be recognised to be of greater value than models which only address visual demands.

Another point of interest is that coupled data and semantics offer the possibility that a project model could include entities related to several design domains, perhaps at several levels of detail, and that tools could work with coherent data subsets for specific decision support tasks. Support for concurrent mixed resolutions in a data model is tightly coupled to semantics and design intent. This has been explored in the Radiance desktop, where for particular visual assessments the glazing on external surfaces (in the thermal context) is subdivided (for the visual context). Another example is the use of thermally passive elements (e.g. obstructions) to support mixed integrated thermal and lighting assessments. In practice, the simulation data model is evolving to support assessment domains not contemplated in the integrated data model. As such, the Project Manager forms an efficient containment of project data and could conceivably act as the agent in communication with an integrated data store.

The IBDS demonstrated an initial step towards implementing a form of tight binding within a palette of tools. Further work is required to support the frequency of interactions observed within the Project Manager. Given the rapid traverses between CAD and attribution functions observed in the case studies, the separation of CAD and attribution functions is an artifact which imposes a considerable burden on inter-process communications and user attention.

One intermediate development would be to augment the existing consistency manager (shown on the left of Figure 7.15) with blackboard and knowledge base agents(shown on the right of Figure 7.15) which are designed to detect specific patterns of journal messages. Currently the consistency manager updates the data model or invokes technical extensions only on the basis of specific user interactions and events within the Project Manager and only within an immediate context. For example, the dependencies related to the addition of a surface in a zone are easily represented by procedural logic.

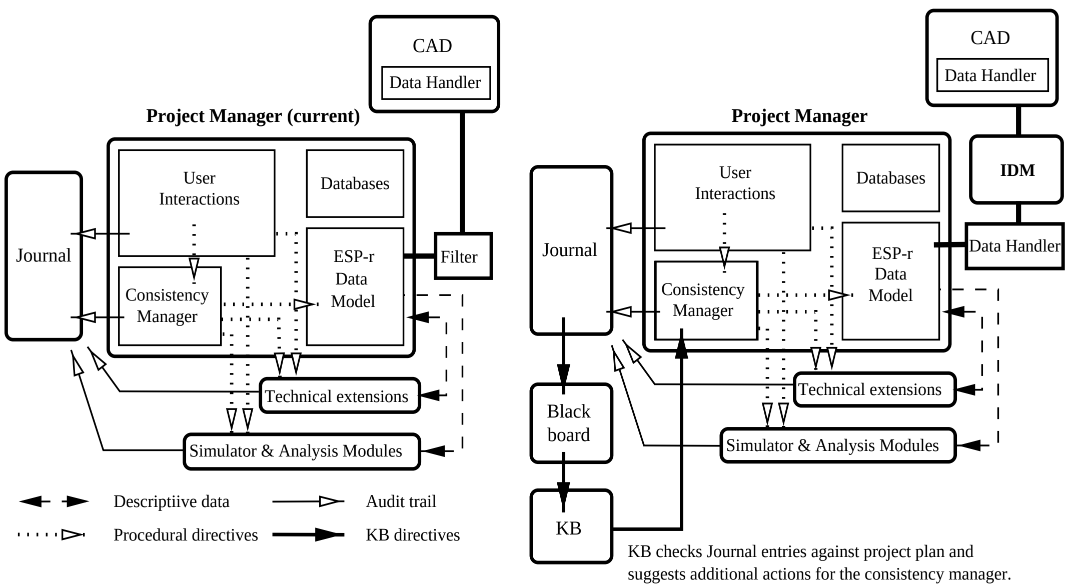


Figure 7.15 Introduction of knowledge-based consistency management.

Detecting that a user is systematically enhancing the resolution of the model (e.g. computationally intensive dependencies should be resolved only when the model is stable), or has progressed a model sufficiently to warrant a QA check, involves a temporal dimension. Detecting that a user's models consistently fault during shading analysis requires that the consistency manager reacts to events outwith the Project Manager. There are also combinations of actions which may each be syntactically correct but which will have unintended or pernicious consequences if carried out in a particular order. An example would be two models (say a base case and reference) which share many common details and where a revision intended only for the reference model is also applied to the base case. A knowledge base which included rules about what actions cause problems in projects with model variants could then request confirmation from the user.

Such a mechanism, by implementing one aspect of deep control, would both enhance the consistency of models and the user's ability to rapidly evolve model variants. As such, simulation would then be better placed to take part in exchanges with an integrated data model and to act as a platform for further explorations of formal process models and performance assessment methods.

In terms of the current work, the use of message passing conventions also opens the way for a better understanding of how simulation tools are used to deliver design decision support. The information contained in journals is of interest to those who develop simulation tools, those who manage simulation projects and those who wish to understand skills acquisition. This is discussed in Section 8.7.

7.9 The need to work differently

This chapter has described an IBDS which, like the Project Manager is a step forwards in design decision support. The Project Manager supported greater model complexity and reduced the attention demanded of the user, but could not ensure appropriate simulation methods were used. The IBDS allowed the design process to be guided by a process model and provided a useful tool for encoding performance assessment methods, if design teams were willing to adopt a different paradigm. The discussion thus returns to skills acquisition and methodology as the remaining barrier to the application of simulation within the design process.

References

Augenbroe G. "Integrated building performance evaluation in the early design stages," *Building and Environment*, Vol. 27, no. 2, pp 149-161, 1992.

Autodesk Ltd *AutoCAD Release 10 Reference Manual*, Autodesk Ltd. Exeter, 1989.

Clarke J.A. *The future of Building Energy Modelling in the UK*, A report to the Building Sub-Committee of SERC, 1987.

Clarke J.A., Mac Randal D. "An Intelligent Front-end for Building Performance Appraisal", *Proc. BEP '91*, Canterbury, April 1991.

Clarke J.A., Hand J.W., Mac Randal D.F., Strachan P.A. *Final Report for the COMBINE II Project*, University of Strathclyde, Glasgow, August 1995.

Clarke J.A., Hand J.W., Mac Randal D.F., Strachan P.A. *Final Report for the COMBINE II Project: Appendices 4 and 5: The ESP-r Aspect Model*, University of Strathclyde, Glasgow, August 1995.

Conforti F. *Inside MicroStation*, OnWorld Press, Santa Fe, 1994.

Javor A. 'Petri-Nets in Simulation', *EUROSIM - Simulation News Europe*, November, pp6-7, 1993.

Johnsen K., Grau K, *TSBI3 Computer program for thermal simulation in buildings User's Guide (Version B08)*, SBI, Danish Building Research Institute, 1994.

Rode C. *Specification of the Generic Tool: BRC, the Building Regulations Compliance Checker*, Danish Building Research Institute, Horsholm Denmark, 1993.

Spiby P. (Ed), 'EXPRESS Language Reference Manual' *ISO TC184/SC4/WG5 Document N14*, 1991.

Ward G. *The RADIANCE 2.3 Imaging System*, University of California, Berkeley, 1993.

Chapter 8

ISSUES RELATING TO THE TRAINING OF SIMULATIONISTS

8 Issues relating to the training of simulationists

> *One marker of well-rounded skills is the point where a user evolves an intuition as to how to construct models. Knowing when the nature of the problem has matured sufficiently to approach the keyboard is the mark of an expert. - Author's synopsis of a dozen workshops.*

This thesis has used a case study approach to identify barriers to skills acquisition by building design professionals and has addressed many of these barriers via the introduction of project management facilities. It has been demonstrated that making tasks straightforward does not necessarily produce better design decision support. Chapter 7 explored the use of knowledge based control of the design process to guide the use of design decision support tools and to detect sequences of user actions which had tactical or strategic implications.

What the Project Manager and the IBDS can not do is ensure that the project methodology and metrics are appropriate, that models are concise representations of the design and that the implications of performance predictions are understood. Some aspects of design decision support remain dependent on issues related to skills acquisition. The development spiral thus comes full circle to the requirement to understand how novices become simulationists, how professionals can evolve methodical approaches to complex assessments, how simulation is actually used in practice and how the design of simulation can support the acquisition of the necessary skills.

8.1 Gathering evidence

This thesis is particularly concerned with how the design of simulation can moderate the resources required for skills acquisition. To better understand how skills are acquired and applied the author introduced facilities to record an audit trail of the user's actions and decision points within an electronic journal. This allows:

- instructors in training sessions to recover the sequence of events leading to a failure in the system or a point of confusion to the user;
- instructors to check if particular simulation topics have been adequately explored and whether participants have fixated on a particular topic;
- managers of projects to determine if error checks have been carried out and the resource required for various simulation tasks;

- developers to track the frequency of use and interaction paths so as to improve the design of simulation tools;

Two forms of journaling have been tested—message passing between design tools and knowledge bases in the COMBINE project, and an in-built audit trail within ESP-r modules. The latter was tested during several training workshops, with individuals undertaking informal tuition at the University of Strathclyde and in support of quality assurance in consulting projects.

The in-built journal records the progress of the user in terms of the decisions they take, the facilities they invoke, the modifications they make to a model, their access to context help and the warnings given to them. Each journal entry is time stamped and given a key word label (to aid in scanning for particular actions). A new journal is created for each design session. By reviewing such journals, it is possible to reconstruct how users reached a particular impasse, discover which tasks are problematic and note which users are failing to take advantage of the support which is available in the tool. The following figures provide examples of how such journals can be interpreted.

The journal in Figure 8.1 indicates that the user: began by scanning training exemplars, selected an exemplar, reviewed the summary and viewed images of its composition and control. The user then ran a simulation, looked at predictions and finally invoked a graphing tool. The elapsed time of the session was 19 minutes. Evidently, the information provided in the exemplar summary was sufficient for the user because they went on to undertake a test simulation without looking at further details of the model.

```
Journal for: student2
Date: Wed Feb 12 16:15:41 1997
PRJ: scanning exemplars enter @ Wed Feb 12 16:15:48 1997
PRJ: owning exemplar @ Wed Feb 12 16:16:00 1997
/export/home/student2/simple/cfg/bld_simple.cfg @ Wed Feb 12 16:16:00 1997
PRJ: current problem @ Wed Feb 12 16:16:02 1997
bld_simple.cfg @ Wed Feb 12 16:16:02 1997
HELP: zones domain summary @ Wed Feb 12 16:16:09 1997
HELP: control domain summary @ Wed Feb 12 16:17:24 1997
PRJ: enter simulation controller @ Wed Feb 12 16:21:14 1997
PRJ: beginning simulation @ Wed Feb 12 16:21:17 1997
PRJ: enter assessment controller @ Wed Feb 12 16:30:48 1997
PRJ: beginning res @ Wed Feb 12 16:30:58 1997
PRJ: enter assessment controller @ Wed Feb 12 16:33:31 1997
PRJ: beginning graphing tool @ Wed Feb 12 16:34:29 1997
Finish project manager @ Wed Feb 12 16:34:46 1997
```

Figure 8.1 Sample Student Journal.

In the session shown in Figure 8.2, the student composed a new problem, beginning with site definitions, then proceeded to create a zone from an extruded floor plan and attributed some of the surfaces in the zone. After the third surface the user found it necessary to leave the zone definition and look at entries in two databases. The user then refocused on the zone and continued attributing surfaces. In the next session the user goes to the databases and adds a new wall type and then uses it in the previously created zone.

The user has assumed that information about wall compositions is available only by going to the database facility. The instructor could advise the user of a reporting option which makes this traverse unnecessary. The tool developer might look to see how often users jump back and forth between geometry and database editing in order to see if a shortcut is needed.

```
Journal for:student1
Date: Wed Feb 12 16:40:33 1997
PRJ: beginning new problem @ Wed Feb 12 16:40:48 1997
HELP: new site @ Wed Feb 12 16:41:44 1997
PRJ: create configuration rideau @ Wed Feb 12 16:41:46 1997
PRJ: new zone @ Wed Feb 12 16:42:05 1997
HELP: extrud @ Wed Feb 12 16:42:13 1997
PRJ: focus on foyer @ Wed Feb 12 16:42:25 1997
PRJ: enter zone vertices @ Wed Feb 12 16:42:52 1997
PRJ: enter zone topology @ Wed Feb 12 16:42:59 1997
PRJ: enter single surface attribution @ Wed Feb 12 16:43:20 1997
PRJ: add def window @ Wed Feb 12 16:46:36 1997
PRJ: enter single surface attribution @ Wed Feb 12 16:49:47 1997
PRJ: enter zone topology @ Wed Feb 12 16:50:09 1997
PRJ: insert surface into another @ Wed Feb 12 16:50:32 1997
HELP: surface details @ Wed Feb 12 16:53:44 1997
PRJ: enter single surface attribution @ Wed Feb 12 16:54:39 1997
PRJ: enter single surface attribution @ Wed Feb 12 17:01:49 1997
PRJ: enter single surface attribution @ Wed Feb 12 17:02:21 1997
PRJ: db management enter @ Wed Feb 12 17:04:04 1997
PRJ: enter primitives @ Wed Feb 12 17:06:22 1997
PRJ: enter composites db @ Wed Feb 12 17:08:36 1997
PRJ: update problem configuration @ Wed Feb 12 17:11:46 1997
PRJ: db management exit @ Wed Feb 12 17:11:46 1997
PRJ: focus on main @ Wed Feb 12 17:15:09 1997
PRJ: enter single surface attribution @ Wed Feb 12 17:15:41 1997
PRJ: enter single surface attribution @ Wed Feb 12 17:15:49 1997
PRJ: add def window @ Wed Feb 12 17:51:14 1997
PRJ: update zone geometry @ Wed Feb 12 17:52:13 1997
Finish project manager @ Wed Feb 12 17:52:52 1997
```

Figure 8.2 Second Student Journal.

Table 8.1 summarises actions from six student journals during a three-day workshop for a mixed group of building physicists and programme managers ranging from novice to expert user.

Table 8.1 Actions from Six Student Journals

Topic / Student	**1**	**2**	**3**	**4**	**5**	**6**
False start	2	2	3	2	7	3
Navigation?	8	5	7	12	8	3
Dialogue?	7	3	4	10	10	2
Pop-up help?	16	6	10	18	16	6
New application?	11	4	11	16	9	1
Sessions	13	17	12	19	10	1
Browse exemplars	9	8	4	11	0	0
Browse database	6	7	5	3	2	1
Shift zone focus	34	23	4	32	38	3
Create zone (in-built)	3	4	1	0	0	7
Import from CAD	2	0	5	13	4	0
Surfaces attributed (singly)	52	21	4	63	68	10

Topic / Student	1	2	3	4	5	6
Global attribution used	0	0	1	29	11	2
View/edit vertices	8	3	0	1	1	8
Check topology	2	1	3	3	3	1
Run simulations	11	4	9	33	8	0

It is clear that Students 1, 4 and 5 were able to perform many more actions during the period. Other students seemed to have stayed longer within particular activities (e.g. learning as they went or performing non-keyboard tasks). Students 1, 4 and 5 had prior experience with simulation tools and were learning how this particular tool worked more than learning its syntactical structure.

The in-built journal is a passive device which relies on expert interpretation and intervention. At the end of a project such journals tend to be discarded. In the IBDS, journals were the mechanism which allowed a set of design tools to function cooperatively. The tagging of messages allowed a number of separate journals to be held and thus the flow of information, requests and responses form a potentially rich source for those who would seek to understand how tools are used to support the design process.

Figure 8.3 shows the transaction monitor used in the IBDS at the midpoint of a design session. Separate logs are kept for transactions (upper section), activity journal (middle section) and application_dialog (lower section) and from this the agents involved in a transaction and their responses can be determined.

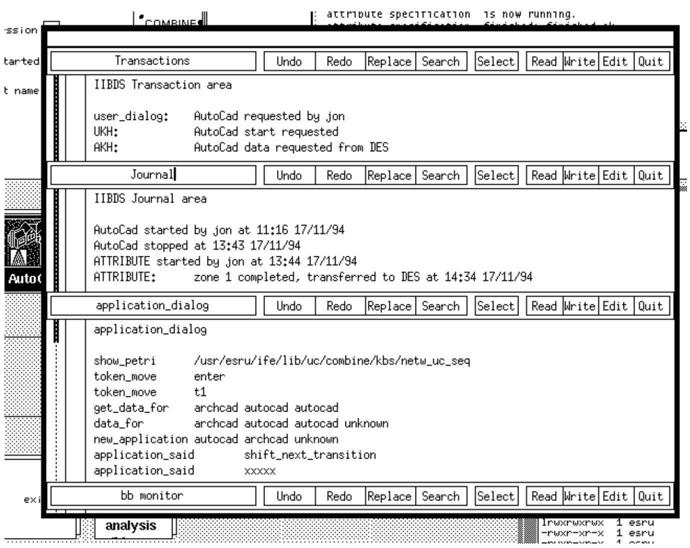


Figure 8.3 Transaction monitor.

The convention of directing messages to various areas of the Blackboard and for knowledge bases to listen to particular areas allows messages related to the *who, when and what* of a design session to be recorded separately. In the fragment in Figure 8.4, the design tool function name, the user name and the time are included. The last line is a message generated from within the attribution tool (Project

Manager) added so that a particular point in the attribution process was recorded (much as the in-built audit trail functions).

```
AutoCad started by jon at 11:16 17/11/94
AutoCad stopped at 13:43 17/11/94
ATTRIBUTE started by jon at 13:44 17/11/94
ATTRIBUTE:  zone 1 completed, transferred to DES at 14:34 17/11/94
```

Figure 8.4 IBDS Journal-area messages.

The IBDS used messaging conventions primarily to coordinate the use of tools rather than to support the understanding of how simulation is used in particular contexts. It is altogether possible that knowledge bases could be designed to address issues of skills acquisition and that journals and audit trails enabled by the current work will lead to new approaches to skills acquisition.

While an assessment of the scores of journals collected over several years is beyond the scope of the current work, a number of trends have become evident. The observed differences in skills acquisition by novices and those with prior experience appear to follow patterns described in the literature [Zuber-Skerritt 1997] and by groups such as Association for Learning Technology (ALT), the Design Research Society and the Instructional Technology Forum who seek to understand the computer’s role in skills acquisition and to provide guidance on the design of courseware and instructional materials which might have application within simulation. The list of references provides contact information for these groups.

The general pattern of those with prior assessment experience has been to make sense of how simulation works by constructing their own internal system (i.e. a mental model or map) which is subject to revision as new information becomes available or new explorations confirm or refute assumptions being tested. They use directed questions to instructors or mentors and directed exploration of the tool to confirm relationships and representations. Successful practitioners continue this process of questioning and exploration as they apply simulation within the design process. This description matches closely that of Constructivism (as defined by Zuber-Skerritt). The observed pattern of skills acquisition also follows that of Kolb’s [1984] cycle of experiential learning shown in Figure 8.5.

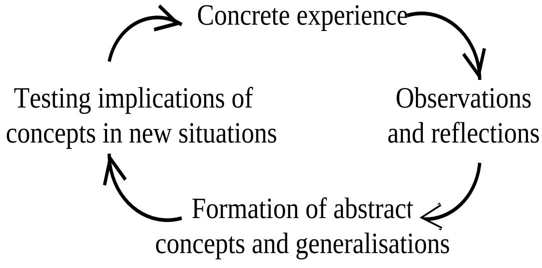


Figure 8.5 Kolb’s experiential learning model.

The observed differences between those with prior experience and undergraduate students also fit the differences between andragogy (the science of adult learning [Knowles 1985]) and pedagogy (the science of early learning). Those with prior experience tend to follow the pattern of self-directed learners, while undergraduates, because of their lack of foundation skills, are not well placed to observe and recognise what is being presented on a computer monitor and thus have difficulty in forming abstract concepts which can be tested and used as a basis for further observations.

Even without an exhaustive analysis of the collected journals the observations thus far:

- confirm the validity of workshop and formal instruction based on progressive exercises and the design of exercises to focus attention on specific aspects of simulation;
- provide a useful frame of reference for the selection and content of exemplars and forms of tool feedback which clarify issues for various types of users;
- provide clues to the difficulties experienced by novices and the need for foundation skills;
- indicate that those who have taken part in mentor-based training in active projects are particularly well placed to deliver insights to the design process.

Section 8.2 discusses the need for fundamentals, Sections 8.3 and 8.4 give recommendations for initial and mentor-based training, Section 8.5 identifies issues for distance learning, Section 8.6 reviews how particular aspects of tool design support skills acquisition and Section 8.7 reviews the contributions and limitations of exemplars.

8.2 The need for fundamentals

> *"If the use of new technology [in teaching] were to begin with an analysis of what students need, instead of an analysis of what the technology can offer, the directions taken would be very different. Every discipline has its fundamental concepts that, being the product of years of research and hard thinking, are necessarily sophisticated. We know from research on student learning that such concepts are often counter-intuitive, or too complex to be understood easily. These ideas may form the central plank in the curriculum for that discipline, and yet remain misunderstood by many students, even after years of study."*
> *Diana Laurillard, in Reinvent the steering wheel, Alt-N Newsletter No 6 July 1994.*

There are aspects of Joseph Heller's *Catch 22* in placing simulation in the hands of a novice who has little or no concept of heat and mass transfer and who lacks the observational skills of a professional. Some are particularly nimble fingered and are able to master navigation skills quickly. It is then a shock for them to flounder as they attempt to proceed beyond set-piece exercises. Such "brick walls" are inevitable until such time as the novice grasps the relation between the underlying physics and the tool's facilities. Indeed why should the novice know there is an underlying representation when the calculation methods traditionally used by the design professions rarely employ explicit representations. Observations clearly showed the tendency of novices to "skim" over the surface of the tool without comprehending that the displayed entities had links to the underlying physics [Crawley and Hand 1997].

The opinions of users as to the importance of methodological approaches and concise models appear to have some correlation with their underlying knowledge of the thermophysical nature of their designs. Practitioners whose experience was with highly abstracted, decoupled and/or steady-state representations of designs were disadvantaged in this regard.

Despite considerable advances in interfaces and ease-of-use, observations of attempts by students to use simulation as a learning tool and of misunderstanding by novices of simulation suggests that there is a minimal set of concepts (i.e. a foundation level) which are prerequisite to taking part in a simulation based course.

First, the thermophysical processes within even seemingly simple designs are invariably complex. Many novices do not believe this. Nor do they appreciate that what *changes*, from design to design, is the *relative importance* of various flow paths and their interconnections. Thus, they tend to accept initial predictions and have little or no concept of model calibration or the need to confirm predictions based on results at a finer level of detail. Secondly, novices miss the essential differences between the visual domain of CAD and the attributed physical domain of simulation and thus they interpret the (often) crude geometry associated with thermal models as they would the geometry of CAD tools and are inevitably disappointed. Lastly, the screen presentations which tool developers and experts easily interpret are often a visual cacophony to the novice.

To encourage appropriate work practices and prevent overly tedious performance assessments, it is necessary for both managers and staff to understand the implications of:

- translating design questions into assessment tasks which can be tested against specific metrics, e.g. "the frequency of temperatures over 24°C";
- the essence of the design and how this can be concisely represented in the syntax of the simulation tool to support assessment tasks;
- quality assurance procedures which are able to cope with models of considerable complexity and which evolve over time.

Such generic issues invariably colour the use of simulation. Thus, initial training focused on keyboard skills suffers from being received as rote patterns with little meaningful content or context, and much scope for misunderstanding. To avoid this, it is essential that fundamental simulation concepts be the foundation on which a tool-specific course is built. Indeed, there are arguments, explored elsewhere [Crawley and Hand 1997, 1998] for simulation fundamentals and generic concepts to be presented in a separate course prior to enrolment in a simulation workshop. The following topics are suggested for such a course:

- classes of assessment tools and aspects of design performance which they are able to deal with;
- general relationships between physical aspects of buildings and environmental systems and their embodiment as "data models" within tools;
- types of relationships between entities (i.e. between site parameters and climatic data);
- how the levels of detail and attribution in simulation models and visual models (e.g. CAD, virtual reality) differ;
- how simulation tool interfaces differ from CAD tool interfaces;
- how to recognise the underlying data model in interface elements.

Even with carefully documented models, the observational skills of novices may not allow them to associate what they see on a computer monitor with physical objects, much less with the thermal interactions within their environment. For example, the relationships between adjacent rooms in terms of geometric representations of the thickness of partitions, boundary conditions and how openings are treated, are implicit in a score of example problems but are rarely recognised by novices.

Nowhere is this more apparent than when users, who think themselves proficient in working with existing models, come up against a mental "brick wall" when trying to compose a new model. An inability to sketch a proposed model on paper is indicative of a lack of underlying skills, ill-formed opinions as to the thermophysical nature of the design or a substantially different concept of what constitutes a model.

A foundation course should aim to enhance observational skills. The sequence of instructions in Figure 8.6 has been designed to reinforce this skill.

• Start with the context of the problem. Where is it located, what are the attributes related to the site that you could find in the tool? Are such attributes presented together or multiple displays?

• Look in the model for documentation. If the author of the model has included a description of what the model is trying to represent then see how much of the explanation you understand. Any confusion you might have should be the subject of questions to the instructor.

• Next, look at the geometry of the problem. What is the relationship between what you see and the "building blocks" of models introduced in the course? How has the building been subdivided into zones? Do these zones match rooms or floors in the building?

• With this as a background, run simulations for a few days in winter and summer and look at the predictions. Are the patterns of room temperatures or heating demands as expected? If not, how might you attain further information or clarification? Are there alternative ways of looking at performance, different levels of detail or even different performance criteria?

Figure 8.6 Instructions reinforcing observational skills

Such a progression of questions and checklists can remind the novice to shift attention occasionally away from the rote learning of keystrokes, to allow them to begin sorting out the thermophysical relationships within and between data model entities. It encourages novices to discover the rules and definitions from which experts derive considerable power. It also puts pressure on tool vendors to improve the documentation of their example problems and the ability of their training staff to support questions not related to keyboard skills, and to implement conventions of interaction which lessen the need for syntactic focused training sessions.

It is recommended that simulation foundations and methods be taught as distinct subjects which are a prerequisite to courses or workshops focusing on specific tools. Formal introductions to methods, if widely taught would help to codify the vocabulary of simulation and enhance the clarity in the exchange of ideas and mobility of skills within the community.

8.3 Recommendations for initial training

The design of instructional materials should account for the user's knowledge of the task which is being undertaken, his or her knowledge of the tool's data model, and how such entities are manipulated within the interface. Table 8.2 includes a synopsis of each type of instructional material. It does not include the case of the novice who has little or no task knowledge or tool knowledge and who should take part in a fundamentals course.

Table 8.2 Topical depth and breadth in instructional materials.

Task knowledge	Semantic knowledge	Syntax knowledge	Approach
yes	no	no	*introductory tutorials* - start from familiar concepts (some rooms get warm in the afternoon while others remain comfortable), link these to high level tool concepts (if you are interested in finding out why conditions are different in different rooms, then a one zone thermal model does not give you this information) and then reveal syntax (here is where you go to find out the details and use the following sequence of keystrokes to highlight all windows)
yes	yes	no	*command references* - present relationship between how an entity, such as a surface, is presented and manipulated and review the thermophysical context and assumptions related to the entity
yes	yes	some	*pocket references* - a concise syntax reminder

In introductory courses and workshops the instructor should start with an overview of the intent and applicability of the tool and then demonstrate some of the tool's salient features, the nature of the models it deals with and the information which it can provide to the designer.

When participants do face the keyboard, one of their first tasks is to use on-line and hypertext tutorials along with basic workstation and window manager manipulations. Their second task is to grasp the relationships between simulation modules and simulation tasks by taking an example problem, reviewing a few of its attributes and then invoking a simple simulation and recovering indicative results. Next, it is essential to establish links between objects in the user's domain (doors, fans, thermostats, etc.) and the semantics of the tool prior to getting caught up in matters of syntax. Experts who are attempting to learn a new tool tend to follow this pattern.

One instructional technique is to ask participants to choose a place which they know well, say a part of their home or office. Firstly, this allows instructors to ask questions and give feedback on form and composition with which the novices are acquainted. Secondly, the resulting predictions can be tested against the perceptions the user has of the performance of the space. Differences found can lead either to an exploration of the model description (e.g. to allow for more infiltration), or to questioning of assumptions (e.g. on thermal mass or fenestration detail).

Ideally, each workshop participant or student should have access to a workstation. If two workshop participants will be working together, they benefit from sharing a workstation for up to half of the workshop. The instructor should have access to a workstation for setting up demonstrations, correcting participant problems and managing instructional media. It is unsettling for a participant's workstation to be repeatedly "borrowed" for maintenance tasks. The instructor-to-participant and participant-to-workstation ratios in Table 8.3 appear to be sustainable in workshops where the work is self-paced.

Table 8.3 Instructor - participant ratios.

	Instructor : Participant	Participant : Workstation
Introductory Courses	1:6 (minimum) 1:4 (better)	1 : 1 or 2 : 1
Advanced Courses	1:5 (minimum) 1:3 (better)	1 : 1

Demonstrations appear to work best with four to six participants gathered around a workstation so that the instructor can maintain eye contact and pace each presentation. Occasional clustering into small groups is also advantageous where the solution to a common problem can be demonstrated on one of the participant's models.

In terms of workshop pacing, the case studies show that workstation sessions longer than an hour are exhausting and increasingly unproductive. It is counter-productive to extend sessions late into the evening and workshops longer than three days are problematic because of the volume of information participants are asked to absorb. Thus, where users need to gain proficiency quickly, a series of short workshops, with the time between given over to individual exploration and practice, is preferable to a single extended workshop.

8.4 Mentor based training

Description and scope

For firms anticipating serious use of simulation, a particularly powerful mode of skills acquisition is to train staff within the context of current projects with a mentor drawn from experienced staff or recruited from outside the firm. In such "on the job" training, simulation tasks, design issues, information demands, interactions, resources (and risk) are real rather than hypothetical. The evidence for this comes from two projects where the author acted as mentor. The first was within the Graham Hills study described in Section 3.3 and the second involved the deployment of ESP-r within a simulation-based consulting firm.

In both projects, the work began with a discussion of the goals of the project, the nature of the information available, the metrics which could be used and likely scenarios for project evolution. In the Graham Hills project the mentor planned and directed the project and in the second case a joint consensus was reached. As the projects progressed, simulation methods, abstraction and alternative approaches for the current stage of the design and the current simulation tasks were discussed. Thus trainees were exposed to the reasoning as well as the techniques even if the specific task appeared to be insignificant. The constrained time frame of the projects allowed the trainees to explore and experience simulation as a continuum. The inevitable inclusion of design changes allowed the exploration of alternative techniques.

Trainees were assigned a sequence of tasks, progressing from supervised repetitive tasks to those requiring judgement and creativity. By "talking through" the decision and descriptive process, staff gained both high level simulation skills and keyboard/ production skills.

Observations

Training via a mentor within a simulation-based project can result in highly proficient staff who are well placed to carry on independent work. The projects also provided indications of the attributes required of the mentor and criteria for selecting appropriate projects and staff, as well as for the frequency of interaction, the progression of tasks and the nature of quality assurance procedures.

- Regular meetings of the design team reinforce the incremental nature of simulation work, limit the tendency to explore tangential issues and provide the possibility of enhancing the place of assessments in a project.
- Regular appraisal of the building and its occupants allows trainees to observe, and then take part in, information gathering.
- The process works well when staff to be trained have an active interest in the outcome of the project, and management take an active interest in staff progress.
- Specific resource and time limits are a source of stress, but they also act to maintain clarity of purpose and conciseness of modelling.

- Mentoring loses much of its power if the mentor is seen to undertake too much of the work.
- Occasional displays of "magic fingers" can motivate by demonstrating the possibilities of proficiency.

A project of moderate complexity and with diverse simulation goals provides staff with sufficient scope for exploring methodology, problem description techniques and project management. Where a number of individuals are being trained—say a manager, an engineer and a technician—the opportunity exists for a whole range of work practices to be discussed and explored.

Issues arising

Not all potential projects and trainees are candidates for mentor-based training. If a project would require 36 hours for a competent team and only 48 hours are available, there is little scope to accommodate the constrained productivity of trainees and every likelihood that the mentor will end up doing the bulk of the work. A one week project extended to a month is also problematic because it lacks focus and trainees will inevitably be distracted. Staff must be willing to put considerable effort into their training, and management should ensure that staff are not distracted by other activities. The pace and intensity of an active project and the need for clear communication argue against novices taking part. In both projects trainees had previously taken part in training workshops.

The mentor requires the means to observe progress without distracting the trainee from the task at hand. Where more than one trainee is involved the mentor’s attention must be carefully rationed. One technique is to assign tasks which involve an explicit check at the end (e.g. define a zone enclosure) and then jointly discuss and explore the result along with the relevant tool facilities (e.g. reports, error checks). The review allows the mentor time to check the work visually and syntactically.

Training within a project is often carried out with the stipulation that design decision support continues apace without concessions to training activities and with the mentor providing quality assurance and ensuring that deliverables do not suffer. This requires judgement on when to accept the trainee’s pace and when to step in and complete critical tasks (with a running commentary on what is being done and the reasoning behind it). When project deliverables are in jeopardy, the mentor may be forced to take over the project, but even this need not compromise the training if the dialogue is maintained.

Mentoring often extends over several projects. The nature of the issues dealt with changes as proficiency is acquired, new tasks are undertaken and working relationships are formed. In the case of the simulation-based consulting firm, the established relationship between the mentor and trainee allowed email and telephone exchanges to be used in the place of direct contact for some tasks.

An mentor to staff ratio of 1:1 to 1:3 is recommended with the mentor having regular access to a workstation and sufficient time and management cooperation to ensure that training and project goals can both be accommodated. Mentor-based training demands considerable resources. In the case of a mentor being brought into a project, preparatory and wrap-up tasks must be included in resource projections.

Shadowing the work of a trainee implies both simultaneous and asynchronous access to a model. Ensuring project deliverables requires judgement on when to accept the trainee's pace and when to step in and complete critical tasks (with a running commentary on what is being done and the reasoning behind it).

Recommendations for mentoring

The combination of an introductory workshop and project based training with a mentor has been seen to be particularly effective. Current practice is to encourage design firms who wish to use simulation in-house to take a staged approach as follows:

- have at least one staff member and manager participate in an introductory training workshop;
- agree on a simulation-based project to be undertaken with the mentor acting as a consultant to the design firm, with the design firm supplying data and design questions and reviewing the progress of the work;
- undertake a second project with the mentor where trainees observe the process, participate in decision making and undertake some simulation tasks under close supervision;
- undertake a third project where the trainees carry out most of the simulation work;
- thereafter projects are supported on an ad hoc basis (e.g. telephone, fax, email, exchange of models).

This sequence allows the design firm to discover progressively the resources and skills needed to establish an in-house simulation group and whether the simulation tool is appropriate. Usually it is only after the second project that the design firm is in a position to judge whether an in-house provision is justified.

The joint participation in a series of projects also ensures that a variety of design approaches can be explored. Projects should be jointly agreed and should, if possible, progress from the straightforward to the complex.

8.5 Distance Learning

For many, distance learning involves access to a workstation, manual and time to 'sort things out' by trial and error. Such a regime is almost always frustrating and misses out on the rich exchange of ideas which typifies other instructional forms. In contrast, informal tuition (self study with intermittent face-to-face access to a mentor) can be quite successful. The question is how to approximate the exchanges observed in informal tuition.

Since 1992, there has been an email based messaging facility (esp-r@strath.ac.uk) for those interested in ESP-r. This has been used by practitioners searching for supporting data as well as a channel for tips and update information. Electronic exchanges are often sufficient to maintain existing relationships between colleagues and between a trainee and mentor. Unfortunately, the considered exchange of information seen in informal tuition is difficult to support via email because it is an inefficient mechanism for building up the rapport needed for mentoring.

Web-based skills acquisition presumes a degree of contact between those seeking to understand and use simulation and the expert acting as instructor. As simulation tools emerge from research environments and begin to be taken up by a larger, and potentially remote, audience, this requirement becomes a barrier.

The current set of hypertext based exercises has been used for remote instruction with email for instructions and progress reports, and file transfers to allow the instructor to review and comment on models. Such links are usually established only after an introductory workshop. The burden of remote communication and the risk of misunderstanding inevitably extends the time requirements for both the instructor and the trainee in comparison with a face-to-face regime. It is also the case that instructional materials tend to be written with the assumption of access (even if infrequent) to an expert. It remains to be seen if those who have little or no previous experience in simulation (e.g. have not taken part in a foundations course) would have much success with remote learning.

Among other things, remote learning facilities lack a remote "observer" to track progress at a detailed level. The user is not often in a position to recount the steps by which they arrived at an impasse. It is difficult to act as a mentor when the patterns of tool use and interaction are not available. Sections 8.5 to 8.8 address some of the constraints of remote learning.

8.6 Tool design in support of skills acquisition

The process of constructing a mental model of a simulation tool and a mental mapping between the user’s expectations of performance and that predicted by the tool requires both systematic and opportunistic traverses of simulation facilities and example problems. It is difficult to short-circuit this process, but the design of tools can both support the "what", "where" and "why" questions which arise. As with other aspects of simulation work, clarity and consistency are essential attributes. As interface elements are found to be consistent they cease to be an issue and the user’s focus can turn to new issues. Tight binding of the interface to the data model, cooperative use of graphic and text attributes and multiple views of performance data are each useful mechanisms for those who seek to understand entities and relationships.

Access to context-sensitive help and tutorials is also useful if users form the habit of using them (the key word being *if*). As always, users’ trust is fragile and a few "no information available" messages can result in such facilities being under-utilised. The need to balance brevity with useful detail has been addressed by incorporating an "additional information" button on help displays.

Where simulation presents difficulties for both the seasoned practitioner and the novice is in the lack of clarity in the link between model details and performance feedback. Too often the number of steps between changing an aspect of the model and viewing the performance implications clouds the relationship. This is inevitable in generic facilities which allow explicit representations of design details and support multiple views of performance.

It remains the case that the provision of generic descriptive and reporting facilities is a barrier for the novice and constrains the use of simulation as an instructional tool. Yet such facilities are indispensable aids for those cases where changes to a model alters performance in unexpected ways and investigations begin to take on aspects of a Conan Doyle novel. The causal chaining of energy balances (see Figure 4.1) is an example.

8.7 The contributions and limitations of exemplars

Understanding of the tool is one aspect of skills acquisition. This must be balanced by an understanding of how the composition of models follows on from the methods and metrics of a project. In this regard, the concept of exemplars is one of the more important deliverables of the current work. The initial concept (see Section 5.1.6) of a well documented, structured and consistent model which could be easily distributed and understood by others proved a defining metaphor for all models. Thus exemplars influenced the design of simulation facilities and led to extensions to the data model.

The initial focus on models which support training (and there are more than a score of progressive exemplars linked to the standard training exercises in Appendix D) has been expanded to form a standard feature of simulation in professional practice. The mechanism for providing access to a hierarchy of training models also allows all of a firm’s past projects to be on-line.

Exemplars have been found to be excellent vehicles for those who have prior experience. However, they have provided uneven support for novices. Several possible explanations exist for this difference. Experienced users are comfortable with the concept of simulation models and file stores and are able to use pattern matching skills to explore exemplars and find similarities and differences with their existing concepts. Novices have only the interface as a guide. Concepts of using the data model to compose abstract representations of a design are not yet established. Simulation appears as a crude approximation of a CAD tool or a virtual reality tool. Exemplars have a potential but only in cooperation with an agent (e.g. a mentor or instructor) explaining what it is they are seeing. The lack of such agents constrains self-instruction regimes and distance learning and one mechanism which addresses this is discussed in Section 9.2.1.

8.8 Conclusions

The deployment of simulation and its role as a decision support tool for the design process is critically linked to skills acquisition. It has been shown that:

- it is both possible and desirable that the design of simulation tools be informed by demands of skills acquisition;
- there is a need to introduce foundation level courses as a prerequisite for tool based training;
- the benefit of allowing non-experts to learn via browsing a range of exemplars and accessing hypertext-based instructional materials has been tested and proven with several score users over a number of years;
- the introduction of tool generated audit trails provides a rich source of information for understanding how users gain skills and deploy simulation;
- those with prior assessment experience approach skills acquisition in a way that is radically different from that of true novices and this should inform future instructional media;
- attaining proficiency sufficient to employ simulation in support of the design process requires a mix of workshop sessions and mentor-supported training in design projects.

References

Association for Learning Technology <http://warwick.ac.uk/alt-E/>

Design Research Society <http://www.mailbase.ac.uk/ lists-a-e/design-research>.

Hand J, Crawley D., "Forget the Tool When Training New Simulation Users" Proc. BS '97, Prague, Vol II pp39-45, September 8-10 1997.

Hand J, Crawley D. "Forget the Tool in Simulation Training", *Building performance*, Building Environmental Performance Analysis Club (BEPAC), Issue 1, pp3-5, spring 1998.

ITFORUM (Instructional Technology Forum) <listserv@uga.cc.uga.edu> an email based discussion group focusing on the design of instructional materials.

Knowles M.S. *Andragogy in Action*, Jossey-Bass, San Francisco, 1985.

Kolb D.A. *Experiential Learning, Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

Zuber-Skerritt O. *Professional Development in Higher Education: A theoretical framework for action research*, Kogan Page Limited, London, 1997.

Chapter 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

This thesis has been concerned with removing barriers to the use of simulation within the building design professions. It has used a case study based approach to identify constraints in current generation simulation tools, simulation practice and skills acquisition and as the basis for conjecture and testing.

The context for explorations is the ESP-r simulation environment, which has been under development at the University of Strathclyde for over two decades. The research began with a review of the ESP-r data model, the nature and functionality of the applications in the ESP-r suite and the nature of the descriptive process which confronts those who wish to apply simulation to design projects of realistic complexity. The review showed the convoluted path by which models are created and the attention to detail required, both of which are considerable barriers to the use of simulation in the design process.

Case studies were used to identify the specific nature of the barriers and were drawn from leading-edge European research initiatives, consulting projects, workshops and teaching initiatives. Simulation was seen to be unsympathetic to shortcomings in the user's attention and to the pace and evolving nature of assessment tasks. The case studies also indicated the importance of methodical approaches to simulation tasks, the demands of cooperative working and the artistry involved in the creation of simulation models which are concise representations of a design. The creation of such models was seen to be compromised by a lack of clarity in the tool's expression of the underlying data model and by lacunae in skills acquisition.

These observations led to a specification for simulation which was more in keeping with the needs of design decision support. The specification included the need for simulation to:

- be open to intermittent use, without the need for prior knowledge of models, and present information concisely, unambiguously and in terms which are familiar to the user;
- accommodate the considerable compositional and operational complexity observed in the built environment as well as project information which is critical to the understanding of the model (e.g. metrics, methods, documents, images);
- allow the focus of assessments to evolve over time—e.g. from system capacity assessment to glare discomfort evaluation—and provide support functions to incrementally add, remove and replicate model entities and systematically apply changes to a model;
- recognise that simulation work is often carried out by a team and that managers need to manage tasks, ensure the quality of models and monitor progress (e.g. using audit trails, on-line process

monitors);

- provide guidance on successful tactics for tool use (e.g. directed approaches to parametric studies) and provide alternative views of performance to aid understanding.

The case studies revealed specific issues relating to the skills required by the users of simulation and to the acquisition of skills:

- there is a link between the efficacy of simulation and the user's non-keyboard skills—forming models, choosing appropriate boundary conditions, setting up simulations and interpreting their results;
- the breadth and depth of options can be overwhelming to the novice and many novices lack the observational skills needed to understand what is being presented to them;
- example problems may illustrate syntax but they do little to show the user how to go about solving difficult (i.e. real) problems unless they are deliberately designed for this purpose;
- training within the context of an ongoing project exposes participants to the *process* of simulation (decisions, interactions, assumptions, etc.) so that they emerge with considerable capabilities.

To avoid the trap of producing corrective mechanisms which addressed only symptoms, the research goals were stated in terms of what is required for simulation to support the design process and what might be required to support those who wish to use simulation:

- Firstly, simulation tool use is about evolving models rather than about evolving files. The mechanics of using simulation should not obscure the task of understanding the performance of a design.
- "What if" questions are central to the design process and imply the need to focus rapidly on a particular aspect of the design.
- Practitioners are often required to shift attention between projects or "come back to speed" on a dormant project. To assimilate models rapidly, users are required to become *au fait* with the intent of the model and the assumptions that have been made in its generation.
- The design process is not well served by "bottom-up" approaches where the model becomes viable only when all the pieces of the puzzle are in place. Simulation must support ad hoc, incomplete and evolving designs.
- The need for unambiguous exchange of information within the design process is clear, as is the need for simulation to "tell the story" of the design in terms meaningful to the design team.
- Simulation should be applicable at various stages of a project and act as a conduit and repository for information within the design process; the marginal cost of additional assessments or modifications should be low and perceived to be low.

- Quality assurance is a critical issue in the deployment of simulation and must inform the design of all aspects of a simulation tool.
- A competent practitioner should expect the task of using simulation to support the design process to be possible with a modest investment of time and attention. Simulation supports a *degree of model complexity* which is not in keeping with the complexity of actual design projects and this must be shifted so that realistic projects can be regularly undertaken.

The primary deliverable of the work is the Project Manager application. This controls all aspects of simulation-based design decision support. The design of the Project Manager is founded on the interface between the practitioner and the underlying data model. It deals with such issues as where the user's attention should be, how error and ambiguity can be minimised, and it deals with the complexity associated with support for the design process (e.g. rapid assimilation of models, accommodation of ad hoc modifications, quality assurance). The Project Manager encompasses: tight binding of the interface to the underlying data model; expression of the data model as objects in the user's domain; the containment of information related to all phases of a design project; an un-noticed interface; access to on-line support and a central desktop tool metaphor.

The interaction conventions used in the current work are that:

- the expression, manipulation and selection of simulation entities should be as objects within the user's domain (i.e. a unified expression of form, topology, composition, name and association);
- simulation functions should be accessible via logical commands;
- dialogue with the user should be supported by suitable defaults, context-sensitive help and is presented consistently and in a standard location, with cooperative use of names, attributes and graphics;
- the interface is dynamically updated to reflect the current state of the model.

Such conventions have been seen to alter the nature of skills acquisition as well as the efficacy of simulation use. They allow the complexity of models to increase without an increase in the overall resource required to mount a simulation exercise, provided users are willing to adapt their work practices in ways which support clear and concise models.

The critical issue of quality assurance has been addressed by designing interactions and reports for clarity, consistency and by introducing a contiguity manager to assess the implications of ad hoc changes to models. A further refinement is the inclusion of the metrics of a project within the problem definition and the production of integrated performance views which allow the unintended consequences of design decisions to be grasped.

A preoccupation with the task of creating model geometry has led many in the design and simulation professions to envisage that close integration with CAD tools will solve many of the perceived problems of simulation. The evidence thus far is that the lack of attribution and topological constraints in CAD tools limit the value of CAD models, and that practitioners underestimate the resource required to provide thermophysical, operational, boundary and control attribution to such models. To compensate for this, the Project Manager includes CAD facilities which are tightly bound to the data model and form an integral part of the descriptive process. However, because geometry is but one aspect of a model and one of many attributes of surfaces and zones, such facilities are given equal weighting to those which support the definition of operational and control regimes.

The efficacy of the Project Manager has been tested in a number of contexts (e.g. consulting and research projects) and its use observed for several years. This has shown that the nature and complexity of simulation models have evolved, as has simulation's ability to support integrated performance assessments and to work cooperatively with other design tools. Whereas users were hard-pressed to maintain models of more than a few score surfaces and constrained control and occupancy regimes, the nominal level of complexity is now on the order of a few hundreds of surfaces and models which combine buildings, network flow and coupled (explicit) lighting control are being attempted by users who are in no way experts. The work has also demonstrated the use of integrated views of performance as an alternative to the traditional exchange of reports.

The goal of supporting the transparent transfer of simulation models and providing access to well documented and well structured example problems (exemplars) has largely been achieved. From a point where simulation models tended to be difficult to understand to all but their authors, it has become an unremarkable event for consultants on different continents to work cooperatively or for participants in the first afternoon of a training workshop to commission a series of simulations on a model of which they have had no prior experience.

Through the use of the Project Manager, a new generation has emerged of competent users who have little or no concept of, or interest in, the underlying file structure. Simple models can be composed and assessed without knowledge of application names or the nature of the data store. This is in contrast to an earlier demand for proficiency in the details of the data store and the functionality of each module in the suite. Training, which once was characterised by rote learning of sequences of commands is much less concerned with keyboard skills, navigation and the interpretation of models.

Another deliverable of the work is a visual simulation desktop to support the acquisition of engineering data needed to drive lighting and blind controls and find the relative frequency of visual discomfort. The ease by which novices include basic visual assessments as an extension of their thermal work is in stark contrast with the effort typically required to undertake such work. The visual desktop is based on the cooperative use of thermal and lighting simulation and demonstrated how a unified data model and peer-to-peer control can support fully integrated thermal and lighting assessments. At the time of this writing, it is the only system which provides such functionality.

Looking back over the early case studies, the availability of project management facilities would have mitigated many of the difficulties encountered. Users would have made fewer errors, tedium would have been reduced and less effort would have been required to manage projects. However, absence of tedium is not the same as tasks becoming easy. Misunderstandings of the nature of simulation, confusion as to what constitutes an appropriate simulation model and tendencies towards extemporaneous model creation are symptomatic of users who are not sufficiently in control of the simulation process.

To address this, two approaches have been tested. Firstly, the project management facilities were expressed as a knowledge-based integrated building design system. The second approach focused on the use of audit trails to understand how novices become simulationists, how professionals can develop methodical approaches to complex assessments, and how simulation tools can support the acquisition of such skills.

The integrated building design system provided desktop access to a palette of design tools and was based on a blackboard messaging system which coordinated access to design tool functions (e.g. condensation analysis, glare assessments, model attribution) and access to an integrated data model. The maintenance of data integrity during design sessions was, for many observers, the core benefit. It presents the possibility of selecting the most appropriate tool for a task with the system ensuring a robust exchange of model information. Observations of the system and comments from design professionals indicated that it is a radical departure from traditional uses of design support tools and has considerable implications for design practice.

The integrated building design system introduced an agent into the design process to observe the progress of the work and the current state of the model and to evaluate this against criteria such as a project work plan or model of the design process and, if necessary, impose control on the design process. It provided a mechanism for a manager to explicitly define models of the design process (e.g. match the demands of a particular project to particular staff and design tool functions) and configure the integrated building design system to support the metrics and methods of the project. That the behaviour of a computational support environment can be reconfigured on the basis of a compact set of instructions continues to surprise observers.

The use of message passing conventions opens the way for a better understanding of how simulation tools are used to deliver design decision support. The current work has expanded the audit trail, and the information contained in electronic journals will be of benefit in the further development of simulation tools, the management of simulation projects and further understanding of skills acquisition.

The integrated building design system addresses a number of the problems associated with the cooperative use of tools, and if implemented with current technology it would have advantages over procedural control and bespoke filters. Much of this depends on practitioners being willing to adopt a different paradigm for their use of tools and tool developers adapting tools to work under knowledge based control. Many of the tasks undertaken by the Project Manager would benefit from process control assistance. Such software developments may or may not be forthcoming. But what is certain is that practitioners are not well placed to use either the integrated building design system or Project Manager functionality unless issues of skills acquisition and methodology are dealt with.

Deficiencies in training are expressed in terms of poor quality assurance, chaotic and complex models, missed deadlines and frustration. This was true at the start of the research and continues to be descriptive of users of current software. To put this statement in context, consider that the design of software can alter the shape of the tool learning curve, it can reduce ambiguity, it can provide support for decision making and reduce much of the frustration of interactions. But this addresses only a fraction of the problems associated with providing design decision support.

Observations showed that novices and those with prior assessment experience demonstrate radically different strategies for skills acquisition. The latter are able to use directed explorations of the tool and exemplars to construct mental models of how the entities in the tool's data model can be used to represent aspects of a design while novices have little or no concept of what it is they are being presented with.

The work found that many participants in workshops and formal tutorials lacked essential foundation skills and suggested the introduction of a simulation foundation course dealing with issues generic to all assessment tools.

The concept of exemplars and the mechanisms which allow the casual browsing of models and exploration of how models support specific design questions has been proven for those with some experience in assessment tasks. Exemplars have contributed to the development of a new generation of users who are aware of user documentation and have begun to use entity naming conventions. Additional work is, however, required to improve novices' understanding of exemplars.

The use of hypertext has been seen to support skills acquisition and works well with exemplars and in conjunction with context-sensitive help facilities. Currently these facilities are designed with the assumption that an instructor or mentor is available to guide the process.

One of the more illuminating discoveries has been how those who have a background in building physics or systems design and have experience with one tool are able to use limited interactions with a mentor to achieve proficiency not normally associated with a self-teaching regime.

In terms of enabling practitioners to deliver useful information to the design process, the combination of an introductory workshop and mentor-based training in the context of a series of active design projects shows clear advantages over all other modes of skills acquisition.

The work shows that the efficacy of simulation within the design process is enhanced by:

- recognition of the ad hoc and iterative nature of the design process and the designer's need for early confirmation of performance trends;

- extensions to the simulation data model beyond the description of thermophysical and systems details to contain all aspects of the design process;
- tight binding of the tool interface to the underlying data model, cooperative use of graphic views and logically named attribution to enhance the clarity of models;
- the introduction of project management facilities to coordinate all aspects of simulation work and enable the transparent exchange of simulation models;
- the cooperative working of assessment tools to support integrated assessments of designs of realistic complexity and the increasing use of integrated views of performance;
- the introduction of knowledge based control of the design process and encapsulation of performance assessment methods;
- the use of progressive exercises in formal skills acquisition, mentor-based training for practitioners and access to a range of exemplar models.

Software does not change the inherent complexity of the task of supporting the design process. It is the task of universities to ensure that those who enter the professions have the foundation skills which mentors can then give direction to. Those who employ methodical approaches to model creation and documentation and whose use of simulation is driven by the need to understand the built environment are the true audience (and beneficiaries) of this work.

9.2 Future work

As simulation is deployed within the design process and professional practices acquire the skills to test the limits of simulation, new design issues will arise. If current trends continue, interest in integrated performance assessments will continue apace and conflated assessments will begin to include visualisation and computational fluid dynamics within a design rather than a research context.

If such assessments are to move from a research context towards design practice both the interactions with the user and the underlying data model will require further development. The further integration of icon-based network definitions and the ability to "overlay" additional assessment domains (e.g. sensors, networks, visual entities, plant components, etc.) over the zone geometry would be a first step in this regard.

The existing contiguity manager is only the first step in ensuring that ad hoc changes in models are reconciled and that the implications of design decisions are communicated to the user. Simulation has some way to go to fully implementing the concept of base case and reference case design studies—for example the restoration of the most recent model after a design variant is deemed inappropriate.

Turning from the Project Manager, the evolution of computing platforms and the ubiquity of the Web suggest that the simulation engine will become a background process which is accessed by a range of conventional applications and Web-based agents. The next section speculates on a virtual laboratory which allows the user's view of simulation to be focused on a specific topic.

9.2.1 The virtual laboratory

Environmental engineering and building physics curricula are usually delivered by a mix of lectures, tutorials and laboratories. While this format supports fundamental topics such as radiant exchange, students often have difficulty envisaging the interplay between physical processes. Given the cost, expertise and time associated with mounting experiments in building science laboratories, it is intriguing to consider the possibility of a *virtual laboratory*. In a virtual laboratory, experiments are composed as simulation models and test regimes as sets of boundary conditions and operational logic. Selected state-variables (temperature, pressure, voltage, etc.) become sensors.

This virtual laboratory is set apart from traditional uses of simulation by an emphasis on its use by those who are in the process of acquiring professional skills. Currently, there are few guidelines for such a facility. Whalley [1995] discusses how the internet might be used to extend computer assisted learning (CAL) tools and how this would change the relationship between the lecturer, the CAL developer and the student. Projects such as INTERACT [CTISS 1993] were set up to explore and develop a suite of simulation based courseware for engineering systems. Observations at the University of Strathclyde suggest that the courseware must:

- be configurable so that the user is appropriately directed (e.g. to progressively more difficult tasks);
- provide focused support for explorations (directions, help, references, links to related information, etc.);
- provide rapid feedback so that "what if" questions can be used to support conjecture and testing;
- include in-built communications facilities for the submission of results and journaling of interactions.

The advent of facilities that allow novices to browse sets of exemplars focused on particular topics and commission basic simulations is very much in keeping with a virtual laboratory. However, it has been difficult to support a sequence of "what if" explorations because of the distracting number of steps that are required. Certainly any requirement to evolve a model expands the complexity of the interaction and the focus of the student will inevitably be towards the tool rather than on the change in performance.

An alternative to the use of bespoke exemplars is to move simulation into the background and present the user with a constrained interface. Consider an exploration into the radiant and convective interactions between a hot surface (say a radiator) and its surroundings. This module might contain only a control panel to allow the size and temperature of the radiator to be adjusted and a display showing relevant interactions. Because there are few interactions a minimum of instruction is required, and feedback to the user can be optimised [Bacon 1996].

The apparent simplicity of a constrained interaction approach would clearly enhance the links between parameter changes and performance. Apart from the convention that a constrained interface be matched with a constrained solution technique, there is no essential reason that a virtual laboratory could not be supported by simulation.

An example of simulation functioning in the background is the Glazing Design Support Tool (GDST) developed within the Image project [LESO-PB, EPFL 1998]. This PC based front-end to ESP-r allows changes to glazing and building location for a range of building types. A command file is generated which the Project Manager uses to update existing models and then commissions appropriate assessments. The result is passed back to the GDST in the form of an IPV report.

Figure 8.7 GDST work session (curtesy of LESO-PB, EPFL).

The mechanism which supports simulation as a background engine is the generation of meta-commands which simulation modules act on to produce an Integrated Performance View report. Commands take the form of *change all atrium glass to double glazed clear* or *relocate building to Nice*. The IPV definition sets the boundary conditions, metrics and focus for assessments and dictates how performance is presented. The pattern established (e.g linking the syntax of the directive to the simulation facilities) appears to be extensible towards the needs of particular audiences.

In a virtual laboratory the point of interaction need not be the same machine which undertakes the assessments. Indeed, the application which hosts the interactions could be a Web based script as well as a conventional program.

References

Bacon R. "The effective use of computers in the teaching of physics", *Active Learning*, Vol 4, pp37, 1996.

CTISS Publications, "INTERACT: Interactive engineering teaching and learning project", *The CTISS File*, Vol. 15, p70, 1993.

LESO-PB, EPFL, *Glazing Systems Design Tool*, EPFL, Lausanne, Switzerland, 1998.

Whalley W. "Teaching and learning on the Internet", *Active Learning*, No. 2 pp25, 1995.

Appendix A

ESP-r SOURCE CODE CONVENTIONS

A.1 ESP-r source code structure

> *It has been argued that collaboration in the field of building energy simulation yields significant benefits for individual researchers—or research groups—as well as for the simulation community as a whole. The main benefits are increased efficiency and output, and more rapid developments. This cannot be achieved by some 'enforced' collaboration, but only on the basis of individuals sharing a common goal or belief. This seems to be equally true with respect to a collaboration support structure and control [Hensen et al. 1994].*

This appendix presents an overview of the structure and coding conventions of ESP-r. ESP-r is implemented as a suite of applications which work cooperatively to support the creation and evolution of simulation models, databases and project documentation required to assess the performance of designs of realistic complexity. It is also implemented in such a way that a worldwide community of researchers is able to use it to explore issues related to environmental performance [Hensen et al. 1994]. This cooperative development model has allowed the robustness of the system and its facilities to evolve over time. In particular, the foundation of common code and library functions has allowed a number of research groups and PhD students successfully to support their research agendas.

The structure of the underlying code is extensive and is partitioned to support a number of development goals.

- Common functions which manipulate the data model are held in a common file store.
- Functions related to the interface (e.g. dialogue, line drawing) and low level data reading and list management are held in a library.
- Code specific to each module is held in separate folders and all other functions are imported at compile time.
- Data files which support system use (e.g. default databases, computing environment specifications, tutorials and exemplars) are distributed with the source code and installed as the system is built.

The arrangement of the source code is shown in Figure A.1 with selected folders expanded. Starting from the bottom of the figure are the folders for low level window manager functions and the ESP-r library. Above this is the Project Manager specific code and the common code which is "imported" as required by various modules. The esrucfg folder (about a third from the top) is an example of a support module which derives most of its functionality from common and library functions.

```
src/
  esp-r/
    Install*      Makefile      Readme        bin/          climate/
    databases/    defaults/     dialog/       env/          esrubld/
    esrubps/      esrucgd/      esruclm/      esrucnv/      esructl/
    esrudfs/      esrue2r/      esrugrd/      esruish/      esrumfs/
    esrumld/      esrumrt/      esrunet/      esrupdb/      esrupdf/
    esrupfs/      esruplt/      esrupro/      esrures/      esrutdf/
    esruvew/      esruvww/      include/      manual/       miscel/
    training/     tutorial/
    esrucfg/
      Makefile     cfg.F        edonecon.F
    esrucom/
      cfgrid.F     e3dviews.F   ecasctl.F    econstr.F    econtrol.F   edatabase.F
      egeometry.F  egrid.F      egtgeom.F    emfnetw.F    emkcfg.F     emoist.F
      eroper.F     esru_misc.F  esystem.F    parsepar.F   plelev.F     pltcfg.F
      pltcfg.Ftodo pnread.F     pnwrite.F    psychro.F    pwrsim.F     scdefs.F
      scndot.F     scsys.F      sensa.F      setup.F      sort.F       startpsf.c
      startsfz.c   startsfza.c  starttfz.c   startup.c
    esruprj/
      Makefile    arrow.F     blcond.F    bldsht.F    bnlthp.F    bpfcom.F
      bpfcontrl.F buspwr.F    cadio.F     context.F   ctlwrt.F    edbmisc.F
      edcasctl.F  edcfd.F     edcfg.F     edcon.F     edcondb.F   eddb.F
      edgeo.F     edmrt.F     edobs.F     edoptic.F   edtopol.F   edzone.F
      gtopol.F    hcfmk.F     insert.F    mfprb1.F    mfprb2.F    mk-prj
      prj.F       prjdmds.F   prjfmk.F    pwrprj.F    read3dv.F   rexmpl.F
      subs_in_prj
    lib/
      ww/@        ww_3.9_sun/
      esru/
        Makefile         Makefile_nb      esru_fc.f        esru_fc.f_nt
        esru_low.f       esru_ter.f       esru_ternb.f     esrublk.f
        no_wwlib.c       wwlib.c
```

Figure A.1 Source code layout.

This arrangement is used in each ESP-r development site. In practice, a researcher/programmer interested in extending the functionality of the system will download the entire source code distribution and then install a standard version of the system. When this is complete, a separate set of development folders is established and relevant portions of the code copied into it. Typically it is sufficient to "point" to the standard code archive for folders which are not being altered. After testing and updating of related documentation, exemplars and databases, the revisions are passed to ESRU for incorporation into the ESP-r archive. It is important to note that such revisions are accepted only if numerical methods are well documented and supported by an unambiguous interface and that system documentation and training materials have been updated.

A.2 Links to the underlying machine environment

The design of ESP-r takes advantage of the multi-tasking and multi-user facilities of UNIX® as well as the rich set of utilities associated with that environment. ESP-r assumes a virtual and distributed computing environment—where it is an unremarkable to be drawing computational power from machines in another country or to be working cooperatively with a colleague in New Zealand with similar ease of a colleague 100 metres away. It also assumes the utter security of the system and models (e.g. users cannot corrupt corporate databases, consulting documents may be protected).

Until recently PC based operating systems did not offer multitasking facilities and true multi-user facilities await future developments. The speed of these inexpensive computing platforms is balanced by a general lack of security and absence of sophisticated scripting facilities. Worse, there are numerous barriers working in a virtual mode. It is not (at the time of this writing) straightforward to work remotely or to share data as transparently. The lack of multi-user facilities blurs the differentiation between the administrator of the simulation software and the user.

It continues to be the case that the power of a workstation for design decision support derives not so much from CPU speed, but from the flexibility it offers for approaching simulation tasks and the facilities it offers for a virtual office. It is likely that PC based operating systems will accrue such attributes and the use of ESP-r need not be as constrained.

A.3 Source code conventions

Support for distributed development and the maintenance of an extensive code base (244,850 lines of FORTRAN and C code) requires careful consideration of coding and documentation conventions. The design of code, data structures, file store formats and documentation relates, in the first instance, to the demands identified in Chapter 4 (e.g. what is required for simulation to contribute to design decision support). For example, the design of the file store is constrained not by the system's demand for compact and efficient transfer of machine-readable data but by the requirements of the project archivist and quality assurance systems.

The design of ESP-r ensures extensive code reuse. For example there is just one function to recover zone geometry and attributions from the file store and this is used throughout ESP-r (it is invoked from 89 points). It supports full error checking and three levels of reporting verbosity. There is also one function which updates the file store with zone geometry and attribution information. However, because the Project Manager owns the data model, this function is invoked only from within the Project Manager and the contiguity checking support module.

The demand for tight binding between the interface and the underlying data model has found expression in all aspects of the code. This, in conjunction with the contiguity manager, requires frequent and distributed exchanges with the file store, and thus the zone update function is invoked at 23 points throughout the Project Manager.

The following figures demonstrate some of the conventions used in the code. Figure A.2 shows a fragment of a zone geometry file (note the annotations) and the code which parses this into the data model. The code proceeds as follows: if the zone type is "GEN" then a line is read (discarding any comments in the line and checking that there are three data items on the line). The subsequent functions parse one item from the line into the data model (the first call recovers the number of vertices in the zone) and test that it is within a given range. Each function includes an error message to display if the process fails or range constraints are exceeded.

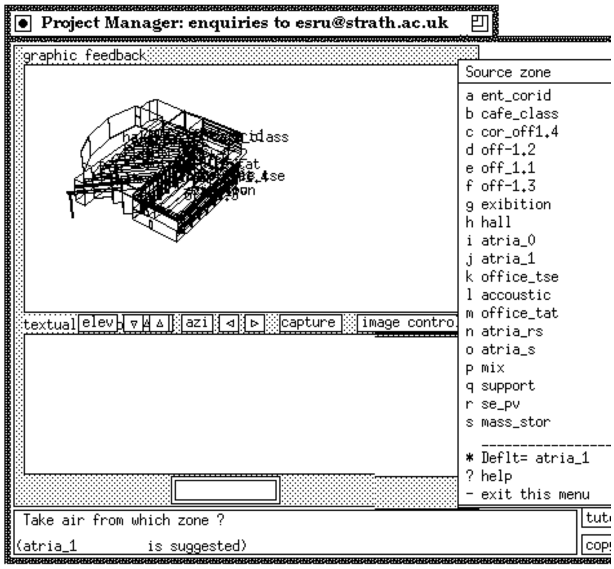
```
[fragment of file store]
# geometry of off_1.1 defined in: ../zone_asis/office_1.1.geo
GEN  off_1.1                # type   zone name
      22      13   0.000    # vertices, surfaces, rotation angle
#  X co-ord, Y co-ord, Z co-ord
     -5.69221     5.69221     3.10000  # vert  1
 . . .

[code fragment which reads the second line]

C Zone is of type GEN.
      ELSEIF(CTYPE.EQ.'GEN')THEN
        CALL STRIPC(IUNIT,ITRU,OUTSTR,3,ND,1,'2nd line NTV NSUR AR',IER)
        IF(IER.NE.0)RETURN
        K=0
        CALL EGETWI(ITRU,OUTSTR,K,NTV,4,MTV,'F','no of vertices',IER)
        CALL EGETWI(ITRU,OUTSTR,K,NSUR,3,MS,'F','no of surfaces',IER)
        NZSUR(ICOMP)=NSUR
        CALL EGETWR(ITRU,OUTSTR,K,AR,-360.,360.,'W','rot angle',IER)
 . . .
```

Figure A.2 Recovery of the data model from the file store.

The user dialogues required to define simulation models tend to be expressed as variants of a few basic interaction types. For example, the definition of inter-zone ventilation requires the user to nominate the source of ventilation. Figure A.3 shows the interaction from the point of view of the user and the code which implements this. The `askzone` function takes the current data value and the list of available choices and provides a prompt and title and presents the list and dialogue to the user. After the user responds, the volume of the source zone and focus zone is used to form the prompts (outs and outs2) for the next user dialog. `EASKR` is the standard call to request a floating point number from the user. The parameters passed to it define the prompts given to the user, the acceptable range of the data and an identifier for use in error messages. The pattern is that each point of user interaction is encapsulated in a single high-level call whose parameters help to document the intent of the code and whose prompts and range checking take into account the current state of the model.

This is not to say that the source code design has reached a point of stability. Inevitably simulation environments contain facilities which, although concise in their numerical expression, are constrained by an inability to secure an equally concise data structure or mode of interaction. The inefficiencies, errors and frustrations which result stand in stark contrast to those facilities with unnoticed interfaces.

An example of this is the definition of control regimes, where the underlying intent is not to constrain the user in terms of what can be sensed, where sensors are located, what operational laws and actuation can be imposed at any point in time, becomes a constraint itself. Initially user interactions were expressed as a sequence of questions and answers with little or no feedback to the user and assumed that all parameters had been arrived at by consultation with a set of tables in the User's Guide. Expressing this as an interactive facility with enhanced feedback and input checking required a fifty-fold increase in coding. Even with this, the use of the facility requires a degree of attention to both interaction and error checking which many practitioners find distracting.




```
 . . .
C Ask the user for source zone and update IVH.
          IV=IVH(ICOMP)
          call askzone(iv,IVH(icomp),' Take air from which zone ?',
     &      'Source zone','Ventilation definition requires a zone.',ier)
          IVH(ICOMP)=IV
          VAL=ACVH(ICOMP)
          ratio = VOL(ICOMP)/VOL(IV)
C Generate prompts and ask user for high range ventilation rate.
          write(outs,'(a,F7.3,a)')
     &      ' High-range ventil. (ac/h where 1ac/h in current =',
     &      ratio,' ac/h in source)'
          write(outs2,'(a,F8.2,a,F8.2,a)') '(vol current=',VOL(icomp),
     &      ' vol source=',VOL(iv),')'
          CALL EASKR(VAL,outs,outs2,0.0,'F',200.0,'W',1.0,'high vent',
     &      IER,7)
          ACVH(ICOMP)=VAL
          TAH(ICOMP)=0.0
 . . .
```

Figure A.3 Zone selection dialogue and code fragment.

Such shortfalls in the support for specific user interactions should be viewed in the context of a balanced approach to tool evolution and the evolution of the data model. Further examples of source code conventions and structure are included within the ESP-r source code (contact esru@strath.ac.uk for instructions on acquiring the source code distribution).

References

Hensen J.L.M., Clarke J.A., Hand J.W., Strachan P. "Joining Forces in Building Energy Simulation" *Proc. Building Simulation '94*, IBPSA, Adelaide, August 1994.

Appendix B
ESP-r DATA MODEL

B.1 ESP-r data model

This Appendix presents a formal data decomposition of the ESP-r system in terms of its building-side, network flow and control data model. It also includes a decomposition of the principal performance (output) parameters. Plant systems, electrical power, computational fluid dynamics, multi-gridding and some of the more esoteric control options have not been decomposed.

The presentation takes the form of ATLIAM diagrams (originally created as part of the COMBINE project [Augenbroe 1992] discussed in Chapter 7) with accompanying commentary. The diagrams have been updated to take into account enhancements to the data model.

Although much of the decomposition of the ESP-r data model is straightforward, ESP-r holds some types of information in two forms (e.g. thermophysical properties associated with each surface and a named attribute which points to an entry in a database) and it is not clear which representation is preferable. Where this is the case, both representations have been included.

The following text gives information on repersentative entities in the decomposition. The ATLIAM diagrams have been grouped into sections as follows: B.2 covers high level descriptions, B.3 describes building and zone entities, B,4 describes geometry, B.5 deals with constructions (thermophysical properties, B.6 operations (occupancy, small power, air flow), B.7 defines schedules used throughout ESP-r, B.8 covers building-side control, B.9 defines output schema, B.10 defines flow networks and B.11 their control. A full listing of the data decomposition is found in [Hand and Strachan 1998].

The diagrams should be understood in the context of the symbols shown in Figure B.1. Entities which are on more than one diagram have the CLONE symbol of a square box. The box between entities is equivalent to Condensation_risk HAS Condensation_details. The arrow between entities indicates that Condensation_risk is a subtype of Performance_assessment_requirements.

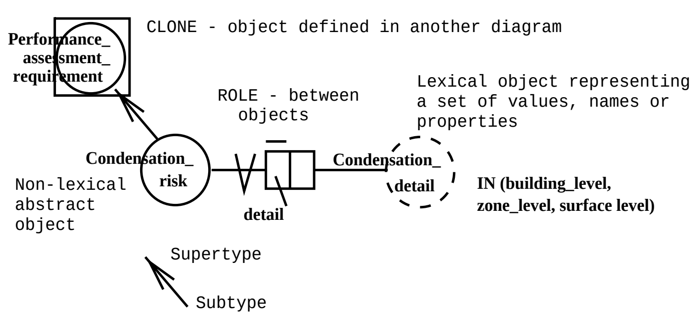


Figure B.1 ATLIAM symbolism.

B.2 High Level Entity Descriptions

B.2.1 ESP_r

ESP_r (Figure B.2) is the top level diagram for the simulation environment. It links the inputs (the Performance_assessment_requirement and the Problem_composition) and uses Modelling_parameter to output (Building_property, Problem_composition, and Performance_assessment).

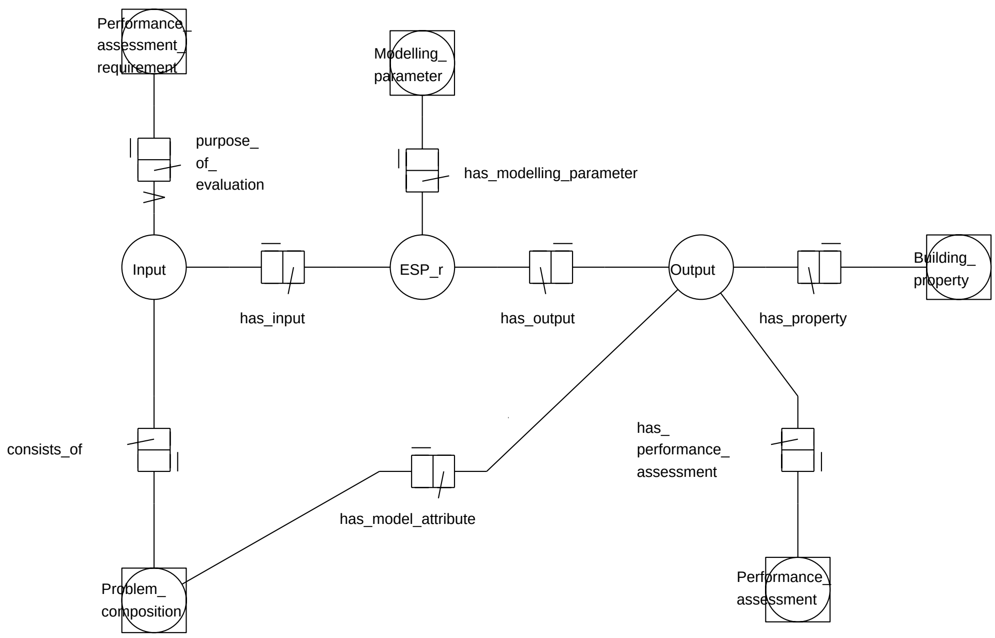


Figure B.2 ESP_r diagram.

Performance_assessment_requirement contains the information necessary for the user to initiate a particular design tool function, i.e. the particular analysis required.

Problem_composition is the definition of the problem, which will be the fully or partially attributed model. After attribution (operational, constructional and control information) it becomes an output.

Modelling_parameter contains simulation-related parameters, e.g. start and stop times of the simulation, start-up simulation period, time-step period and control, alternative algorithms for some of the heat transfer processes, etc.

Performance_assessment contains the primary outputs of ESP-r as a thermal performance evaluation tool.

Building_property contains some of the internally calculated properties of the building which may be of use in other design tools.

B.2.2 ESP_r_problem

The **ESP_r_problem** (Figure B.3) is the top entity for the definition of the problem. It shows how Building, Plant and Flow can be separately or jointly simulated, and gives the context of the problem.

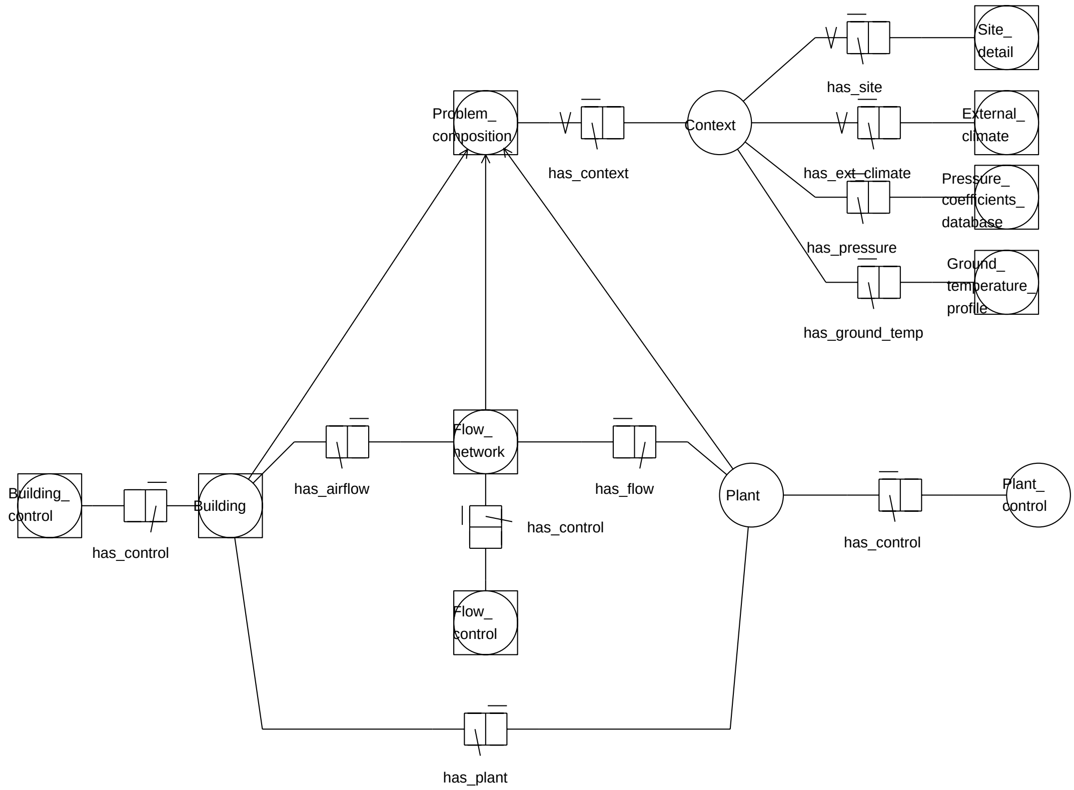


Figure B.3 ESP_r_problem diagram (referenced by: ESP-r, building, flow, plant).

Context defines the boundary conditions in the simulation and comprises Site_detail (building location and its situation), External_climate (meteorological data), Pressure_coefficients (boundary conditions for wind-induced airflow analysis), and Ground_temperature (sets of monthly ground temperatures) .

Building is the highest level entity for building-side simulations, and can optionally have flow, plant, power and moisture networks connected to it. It has associated Building_control which represents the control systems on the building side.

Plant is the highest level entity for plant simulations. A Flow_network and a Building can be associated with a plant system. It has associated Plant_control. Neither Plant or Plant_control have been decomposed.

Flow_network is the highest level entity for flow simulations. Both air and liquid flow networks are possible, connected as appropriate (and optionally) to the building and plant systems. There may be more than one network. There is an associated Flow_control which represents the control systems for fluid flow.

B.2.3 Performance_assessment_requirement

Performance_assessment_requirement (Figure B.4) comprise the information necessary for the ESP-r user to initiate a particular design tool function (i.e. the particular analysis required and the information to be returned). There are six subtypes considered: Energy_balance (relative magnitudes of energy fluxes), Plant_size (capacity assessment), Condensation_risk, Energy_consumption (demand over time), Heating_risk (risk of overheating or underheating) and Comfort.

Associated data are: Zone_name (list of names or "all" for the collection of zones) to be assessed, Dates_of_assessment (period over which the assessment is to be made as a Standard_period or a User_specified_date), Temperature_criterion (range or frequency bin of temperatures), Condensation_detail (search criteria for whole building or a portion of the building) and Comfort search criteria and one or more comfort_measures (PMV, PPD or "comfortable, pleasant", "slightly cool, acceptable", etc).

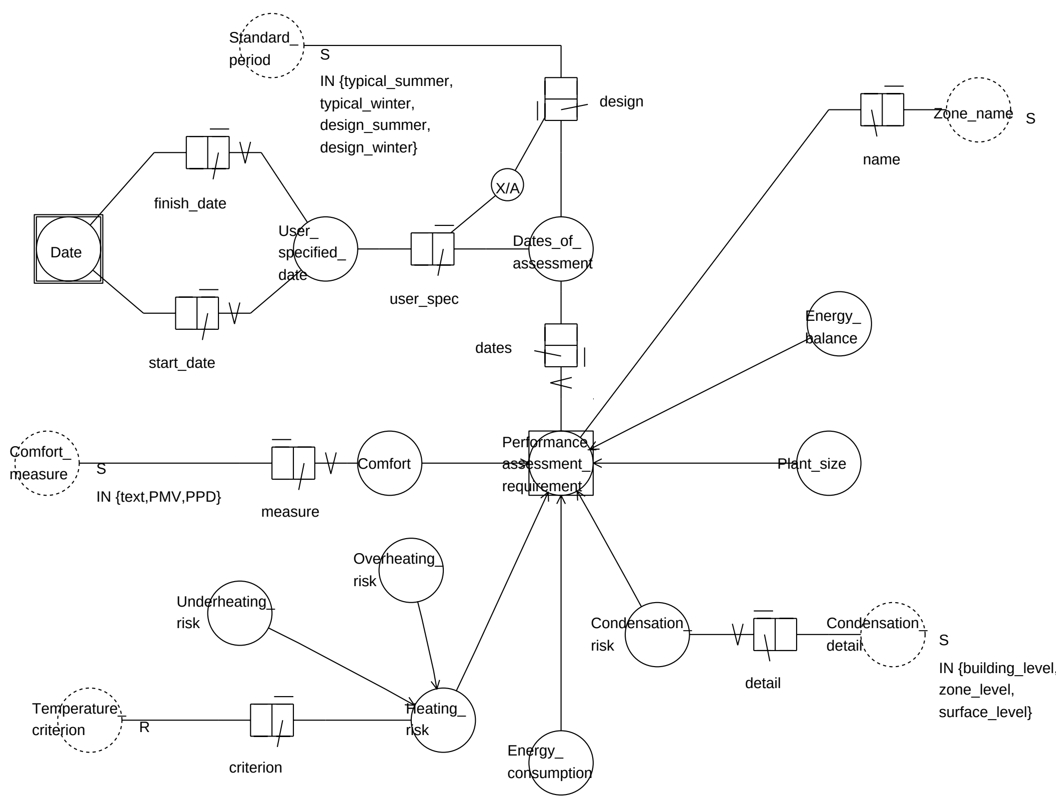


Figure B.4 Performance_requirement diagram (referenced by ESP-r).

B.2.4 External_climate

External_climate (Figure B.5) is the top level entity for meteorological data. It has a start and stop date, usually, but not necessarily January 1st and December 31st. The entities Climate_station_name (site location), Latitude (positive for northern hemisphere) and Longitude_difference (difference from the standard meridian where positive is eastwards) relate to the location of the climate collection station.

Associated data are: Hourly_climate_value(s) which comprise diffuse radiation (on the horizontal plane), external temperature (dry-bulb temperature), wind speed and direction (clockwise from north), relative humidity, plus either direct normal or global horizontal radiation.

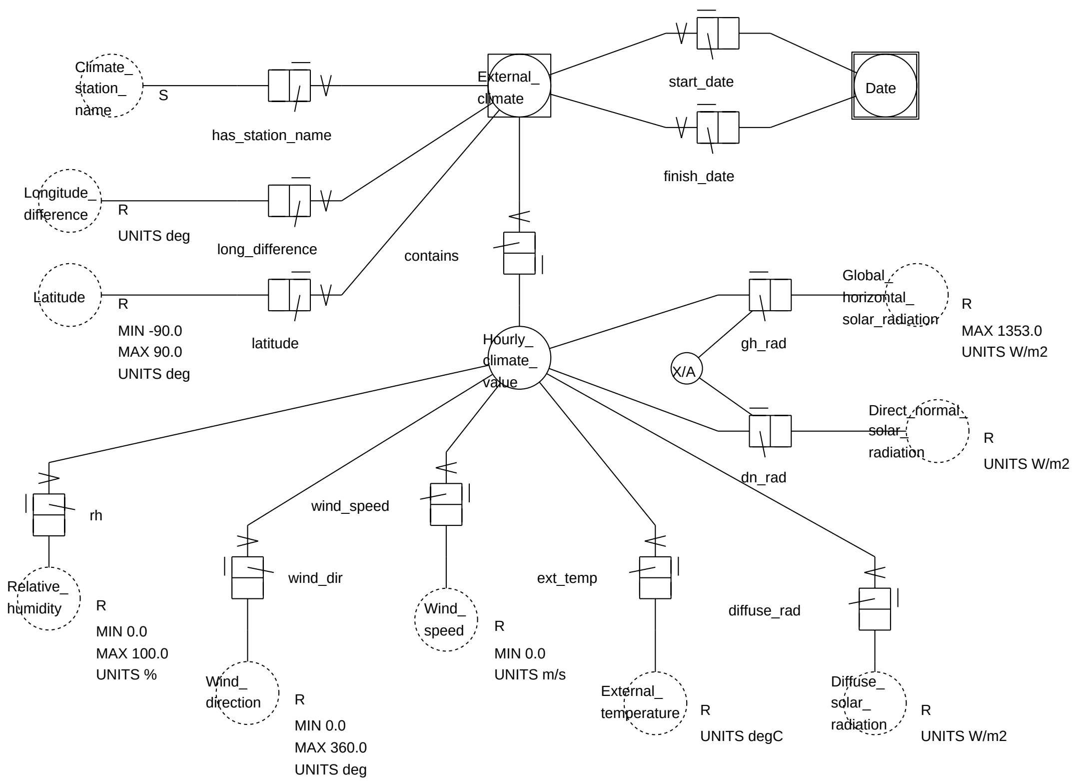


Figure B.5 External_climate diagram (referenced by: ESP_r_problem).

B.2.5 Site_detail

Site_detail (Figure B.6) includes the location of the building, Site_name, a Site_latitude and Longitude_difference, Ground_reflectance (assumed uniform for the site), and a Site_exposure (see Section B.2.4).

Site_exposure relates to the calculation of external longwave radiation, and requires the view factors for buildings, sky and ground. They are assumed to be constant for the building. Location is expressed in terms of a string "urban_normal", "rural_normal", etc., from which an inference is made of the external view factors. Alternatively the user can specify an External_viewfactor_ distribution which comprises view factors to buildings (Building_viewfactor), sky (Sky_viewfactor) and ground (Ground_viewfactor).

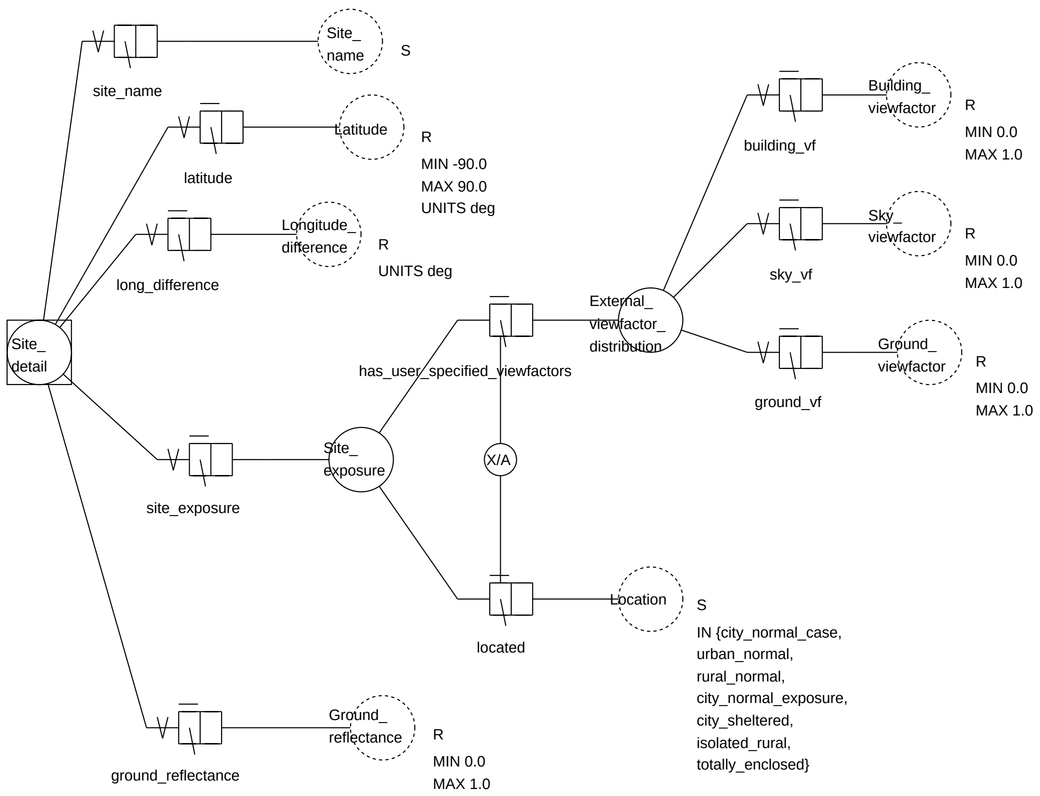


Figure B.6 Site_detail diagram (referenced by: ESP-r_problem).

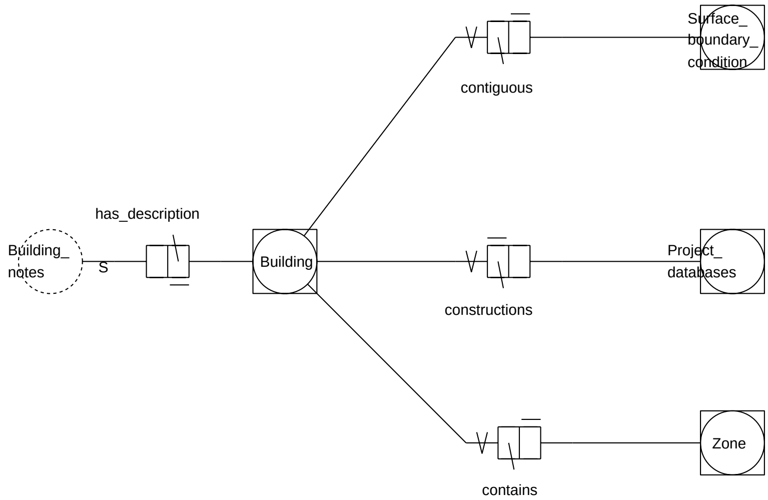


Figure B.7 Building diagram (referenced by: ESP-r_problem, zone, construction_db, contiguity).

B.3 Buildings and Zones

B.3.1 Building

Building (Figure B.7) is the highest level concept on the building side of the simulation. It includes Building_notes (related descriptions and images and project documentation), Surface_boundary_condition (contiguity properties of each surface in the model), Project_databases comprising materials, constructions, optical properties, etc. associated with the building and Zones (fundamental spaces for building simulation).

B.3.2 Contiguity

Contiguity (Figure B.8) must be specified for each surface in each zone. The Surface_boundary_condition entity relates to the building level because it includes contiguity of zones/surfaces and supports consistency checking. It is also a surface attribute of the entity surface (see surface diagram).

Surface_boundary_condition has six mutually exclusive subtypes: External_climate (taken from the selected climate database), Relative_condition (adjacent air temperature is a Relative_air_temperature to the simulated zone), Fixed_condition (specified air temperature and incident radiation_on_surface_boundary), Contiguous_simulated_zone (adjacent conditions are another simulated zone and specified as Contiguous_zone_name and Contiguous_zone_surface_name), Ground_temperature (adjacent to a ground temperature profile), and Adiabatic (no flux transfer across the outermost boundary).

B.3.3 Zone

Zone (Figure B.9) is the fundamental space for building-side thermal simulation. Each zone contains Geometry (3-D geometrical data and references to constructional data) and Operations (details of heat gains from internal heat sources, and a simplified treatment of ventilation and infiltration heat fluxes). There are optional extra features, such as Obstruction_geom (geometric entity which shades surfaces), user-specified Convection_coefficients, and Timestep_data (measured data that can be superimposed on a simulation, either airflow data or heat gains measured time-step data). Lighting_control can also be specified at this level.

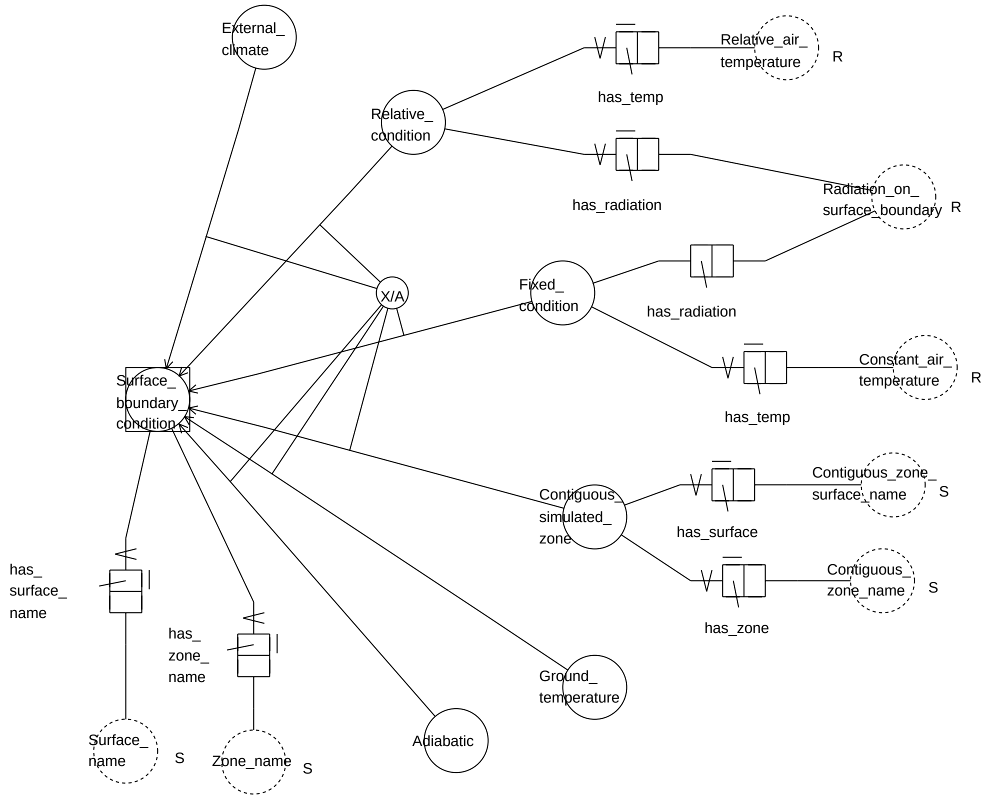


Figure B.8 Contiguity diagram (referenced by: building).

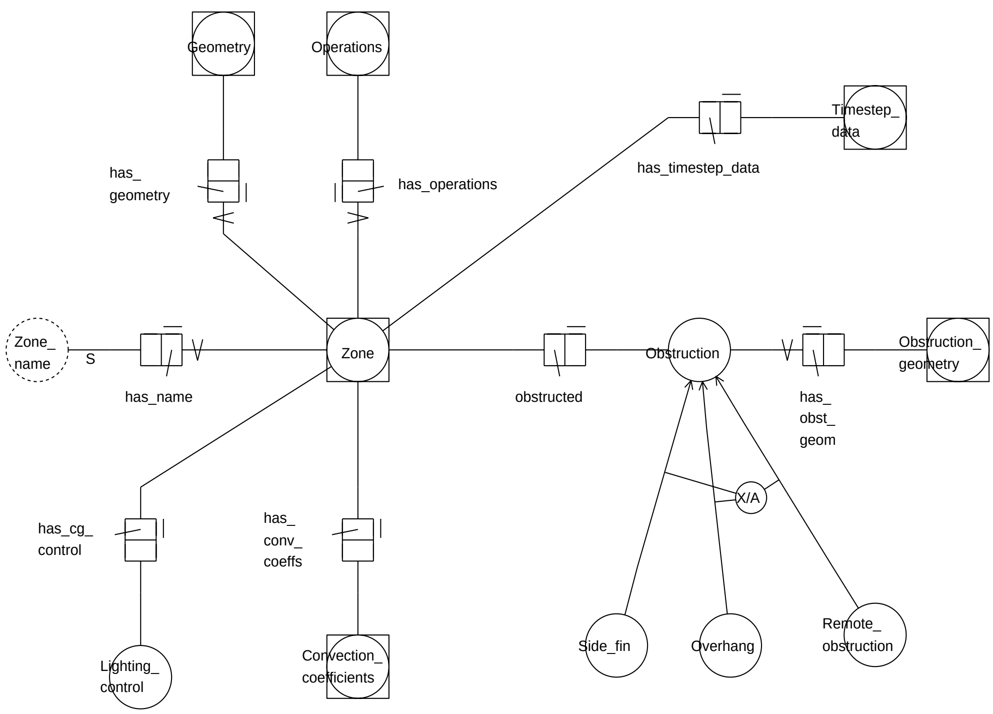


Figure B.9 Zone diagram

(referenced by: building, obstruction_geom, geometry, operations).

B.4 Geometry

B.4.1 Geometry

In ESP-r there are two internal ways of specifying geometry. The basic form (Figure B.10) is a Geometry-shape of subtypes General (GEN) body or Rectangular (REC) bodies (obstructions and visual entities).

Rectangular subtype defines a box. The coordinates of the origin (3D coordinate) and the length (in the x-coordinate direction), width (in the y-coordinate direction), height (in the z-coordinate direction) and orientation (angle to the y-axis from north, anticlockwise positive) are required.

General subtype defines a polygon bounded enclosure. In this case the vertices (x,y, and z coordinates) of the zone are input and linked as an ordered list (anticlockwise looking from outside the zone) to define polygon edges.

Associated data: Bounding_vertex is a linked list of vertex indices which define a surface. The geometry of each zone is composed of a number of surfaces. The Surface entity contains information on attributes related to surfaces.

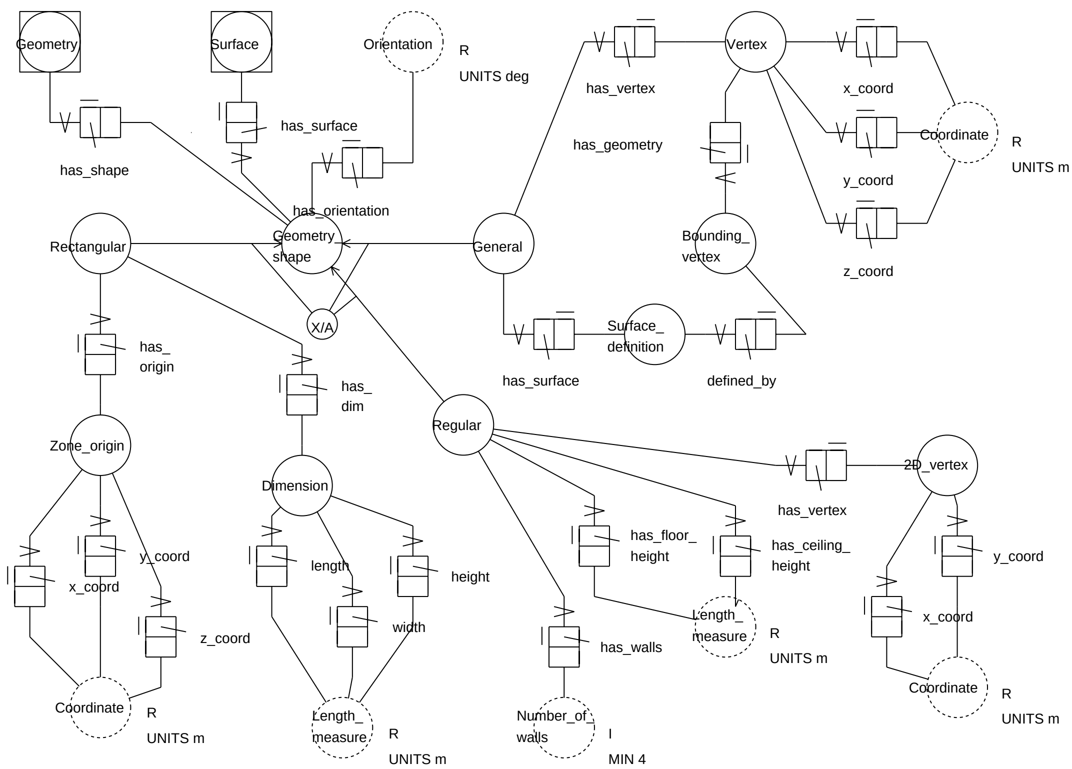


Figure B.10 Geometry diagram (referenced by: zone, surface).

B.4.2 Surface

Surface (Figure B.11) contains information on surface attributes (e.g. Surface_name, Construction_name (reference to the entry in the constructional database) and Boundary_condition) which must be specified.

Boundary_condition supports the variants (exterior, similar, fixed, contiguous_zone, ground, adiabatic).

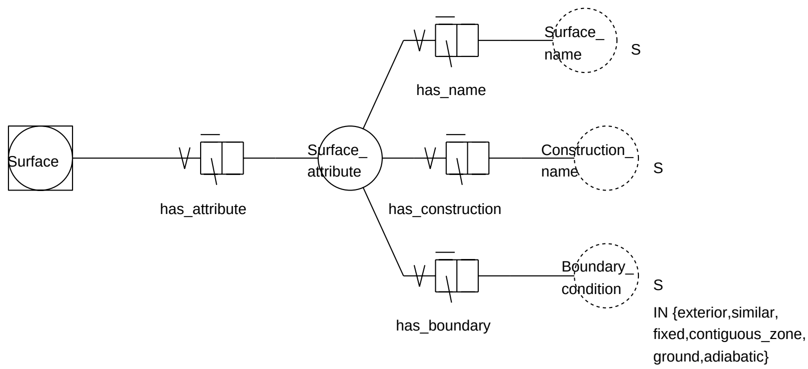


Figure B.11 Surface diagram (referenced by: geometry).

B.5 Construction

Constructional properties have been decomposed using the Construction_database, which contains the thermphysical properties of all constructions within the building. There is also a reference on the surface diagram to the Construction_name (i.e. database entry) attribute associated with each surface.

B.5.1 Construction_database

Construction_database (Figure B.12) is linked to the Building entity, so there is one Construction_database_name for each project. Construction contains the entries in the database.

Materials_database contains the details of the individual Materials (basic thermophysical properties) from which constructions are composed. One Materials_database_name is associated with the project.

Optical_database contains the optical properties for transparent surfaces and is referenced by the Construction_database.

Glazing denotes the optical properties of each type of glazing held in the Optical_database.

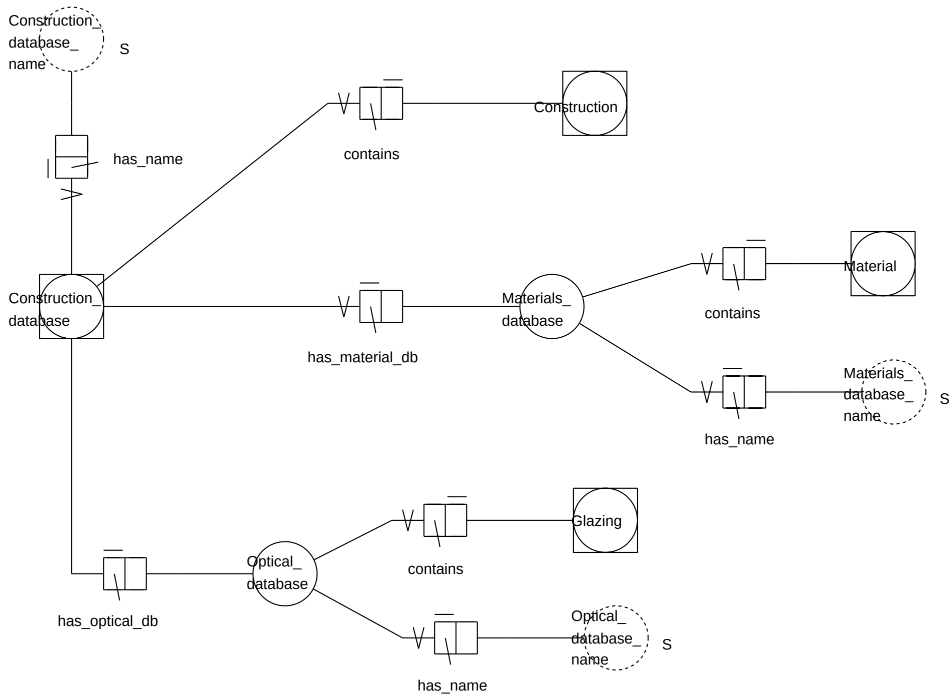


Figure B.12 Construction_database diagram (referenced by: building).

B.5.2 Transparency

As mentioned in the geometry decomposition, glazing systems are surfaces which are given Transparency attribute and have as one attribute a set of optical properties. Optical requirements are the system direct transmission and the solar absorptivity (at several angles of incidence) of each layer in the transparent system obtained from entries in the optical database.

Glazing (Figure B.13) contains standard optical properties (Standard_properties) for transparent constructions. It can also have a schedule for varying glazing properties (Replacement_properties) according to time, incident solar radiation or external temperature. Within the schedule (Glazing_properties_schedule) there is a possibility of control based on some activation level. This is expressed through Standard_properties and Replacement_properties as subtypes of Glazing.

B.6 Operations

Operations in ESP-r are for airflow (an alternative to the network model) and for internal heat gains by occupants, lighting and equipment. Operations_airflow and Operations_casual gains are subtypes shown in Figure B.14.

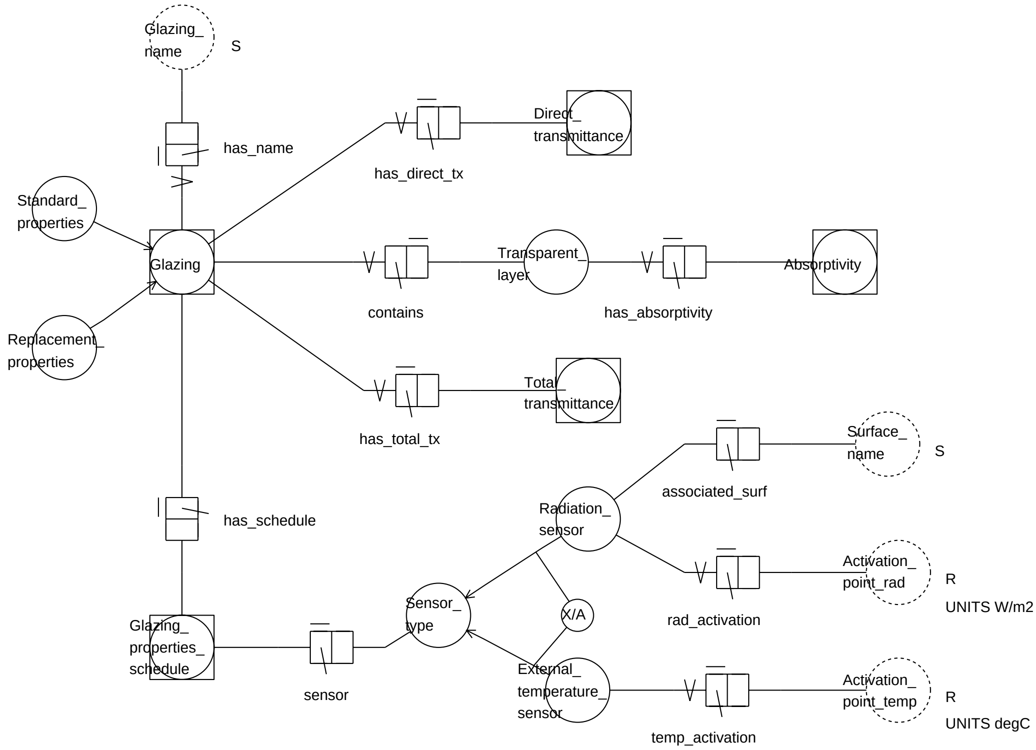


Figure B.13 Glazing diagram (referenced by: constructions, glazing_properties).

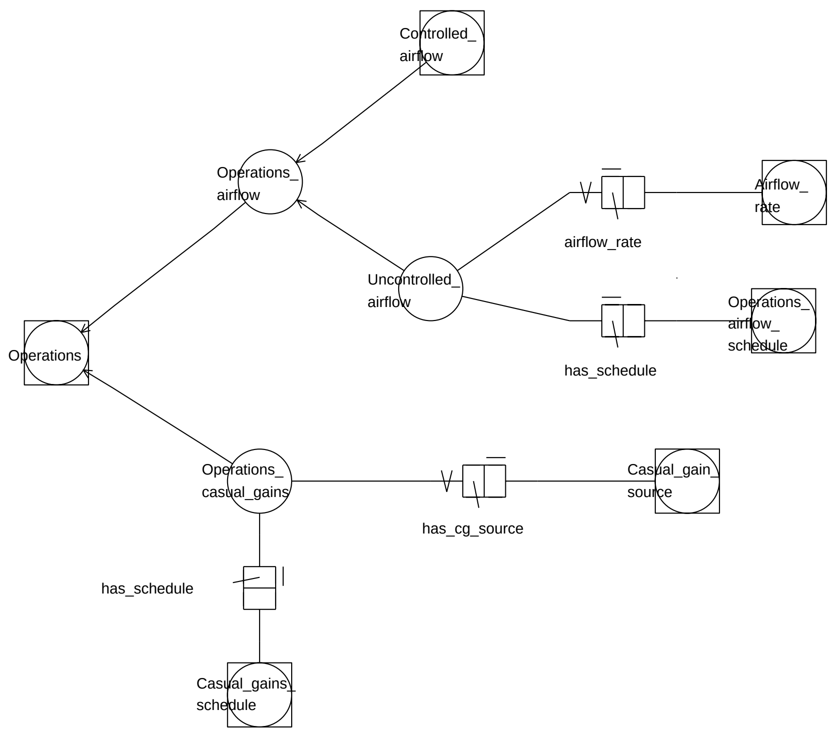


Figure B.14 Operations diagram

(referenced by: zone, casgains_sourec, airflow_rate, controlled_airflow,schedule).

B.6.1 Operations_casual_gains

Operations_casual_gains are composed of one or more Casual_gain_source (Figure B.15). Both can be scheduled differently on weekdays, Saturdays and Sundays. For airflow it is also possible to have thermostatic control based on range of control conditions.

Associated data: Casual_gains_schedule allows a day type (Daytype_wss) to be associated with an Operation. Each casual gain has a convective and radiant fraction (Rad_conv_fraction) the sum of which should not exceed 1.0. In the case of Floor_area_per_occupant, casual gains assume a heat gain per person of 95W sensible, 45W latent.

Casual_gain_source indicates the possible sources (Occupant_heat_gain, Equipment_heat_gain, Lighting_heat_gain, Lighting_heat_gain_per_unit_floor_area, Heat_gain_per_unit_floor_area, Equipment_heat_gain_per_unit_floor_area, Occupant_density) of internal heat gains.

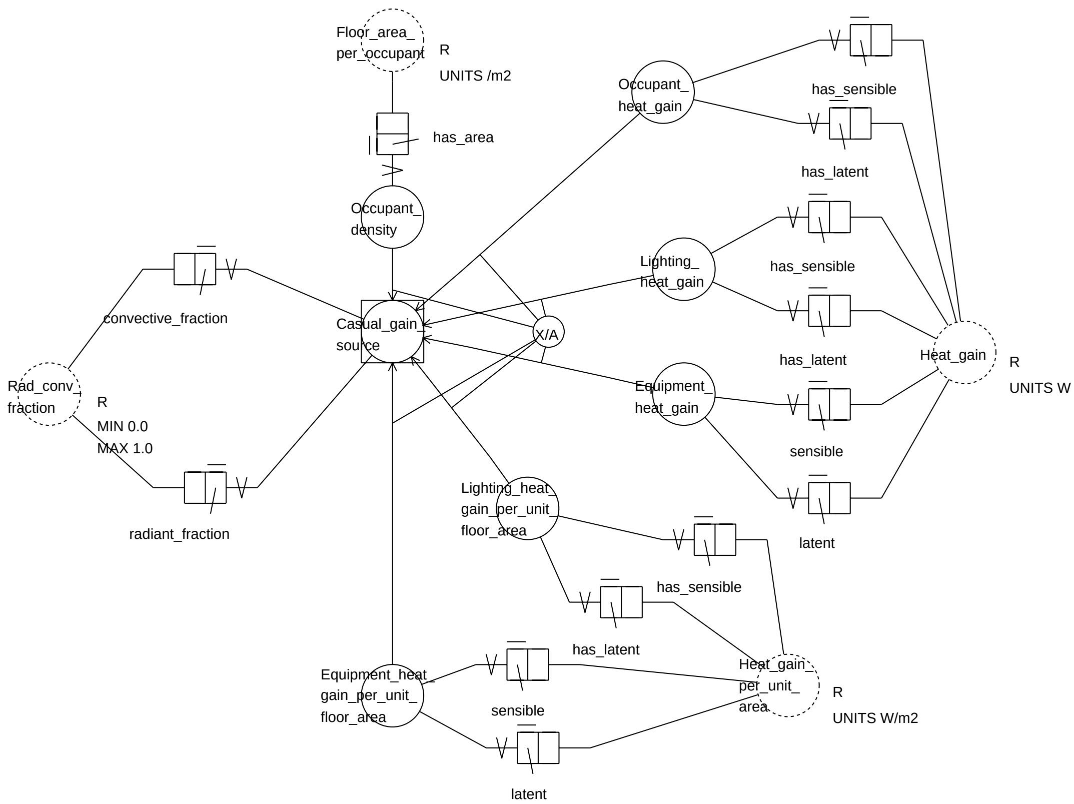


Figure B.15 Casual_gain_source diagram (referenced by: operations).

B.7 Schedule

Scheduling is possible for many of the functions within ESP-r. These have been added to the relevant parts of the decomposition. However, all scheduling has been brought together within the schedule ATLIAM diagram.

B.7.1 Schedule

Schedule (Figure B.16) is a supertype of all schedules in ESP-r. It can have one or more schedule periods which divide up a day. If there are no schedule periods, it is assumed that the control is active for all 24 hours of the day. Current schedule types are: Convection_coefficients_schedule, Glazing_properties_schedule, Operations_airflow_schedule (with the possibility of different controls being possible on weekdays, Saturdays and Sundays).

Building_control_schedule is a subtype which can have dates of validity, as well as day types (weekday, Saturday, Sunday).

Flow_control_schedule is a subtype which offers the same possibilities as building control.

Casual_gains_schedule is a subtype which can have a day type associated, and different controls possible on weekdays, Saturdays and Sundays.

Schedule_period(s) divide up a day, and have one or more start and stop times. The control will be active from the first timestep after the start time to the first timestep after the stop time. To schedule all 24 hours, the start and stop times would be 00.00 and 24.00 respectively.

Dates_of_validity are applicable to building and flow control schedules to allow different building control for different seasons (or indeed different weeks or even days). If not specified control is active for every day of the year.

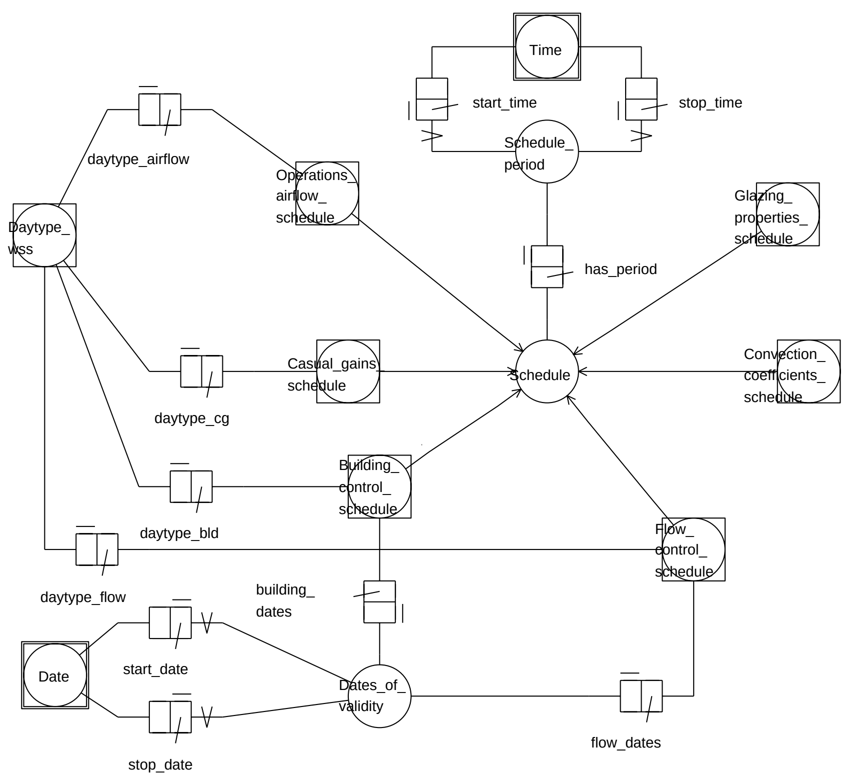


Figure B.16 Schedule diagram.

B.8 Building Control

This section gives details on the building control decomposition. Some of the more esoteric control options have not been decomposed here (see Figure B.32 for all the building control law options).

B.8.1 Building_control

Building_control (Figure B.17) is the highest level diagram for building control. It contains references to Sensor_point (location of the sensor), Actuator_control_point (location of the actuation control point), control types and laws which can be attached to one or more zones. Building_control_schedule is a subtype of schedule (see Section B.7) with the exception that sensor and actuation control points are fixed for any given Building_control.

Control_type determines the properties that are sensed and actuated.

Control_law specifies the control algorithms representing the logic of some, real or imaginary, controller.

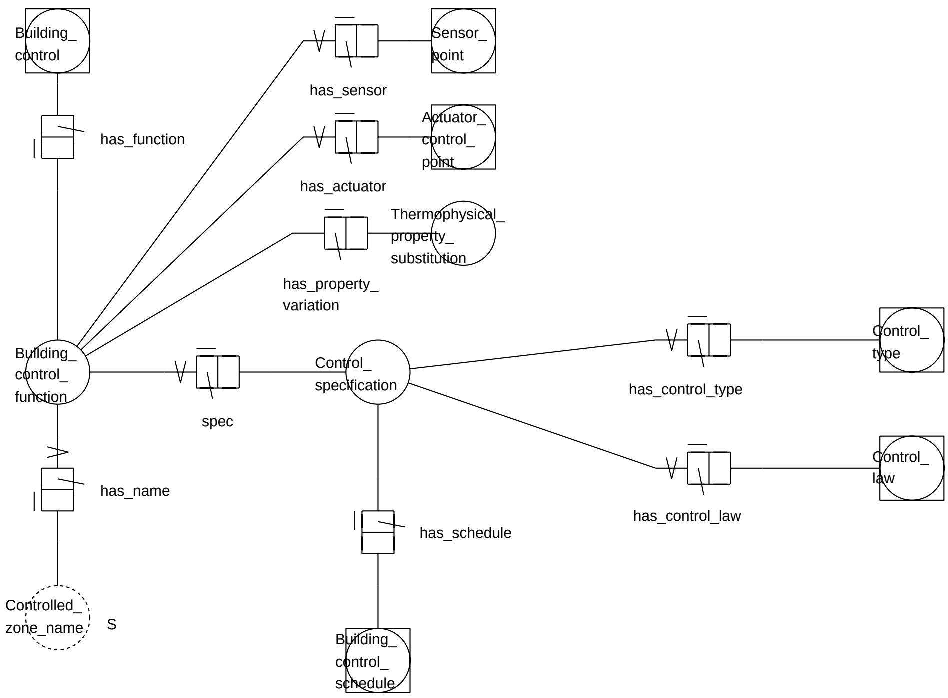


Figure B.17 Building_control diagram (referenced by: ESP_r_problem, schedule).

B.8.2 Control_law

Control_law (Figure B.18) is a superset of the building-side control laws. The more common laws are: Ideal_control, Free_float_control (effectively "no control"), Ideal_preheat_precool, Ideal_fixed_heat_injection, PID_control, Multi_stage_control_with_hysteresis, Variable_supply_temperature_control, Time_proportioning_on_off_control.

Ideal_control causes the sensed condition to attain a specified set-point within the limitations of specified heating and cooling flux limits.

Additional data: Plant_capacity (heating and cooling maximum and minimum values) and Setpoint_temperature (heating and cooling).

Ideal_preheat causes an exponential evolution of the control variable to a specified set-point throughout the scheduled period.

Additional data: Plant_capacity (heating and cooling maximum and minimum values) and Setpoint_temperature (heating and cooling).

Ideal_fixed causes an injection of a specified heating flux as a convective input, if the temperature falls below the heating set point. Cooling is treated similarly.

Additional data: Flux_injection (Fixed power) and Actuation_temperature (heating and cooling).

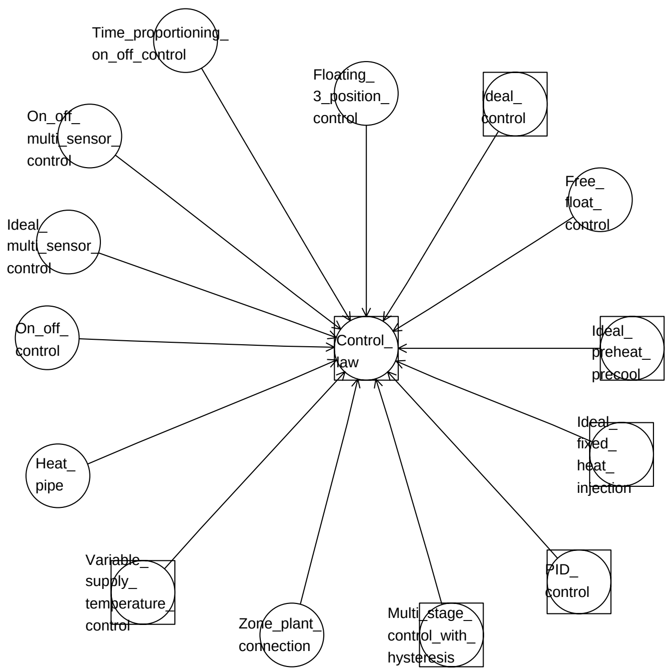


Figure B.18 Control_law diagram (referenced by: building_control).

PID_control (Figure B.19) emulates of PID controllers. Subtypes of PID_control are: P_control (proportional control only), PI_control (proportional and integral control only), PD_control (proportional and differential control only), Full_PID_control (proportional, integral and differential control).

Additional data: Derivative_action_time, Integral_action_time, Plant_capacity (heating and cooling maximum and minimum values), Setpoint_temperature (heating and cooling) and Throttling_range (for proportional control).

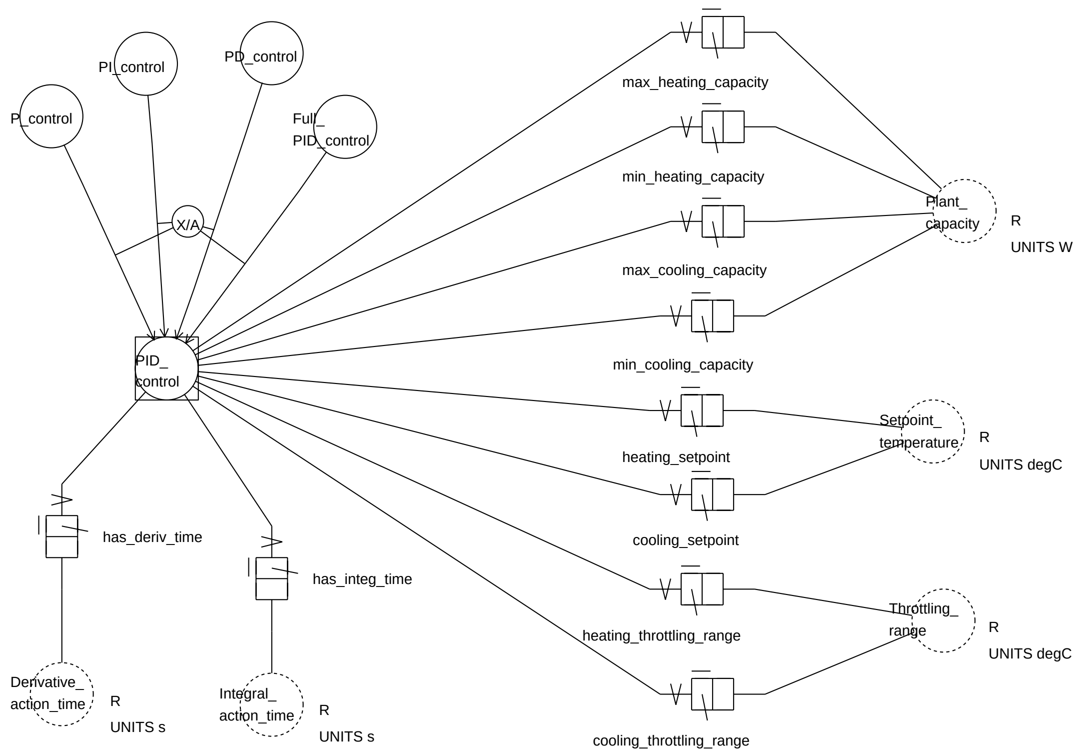


Figure B.19 PID_control diagram (referenced by: control_law).

B.9 Outputs

The possible outputs from ESP-r as a thermal performance evaluation tool can be grouped into building properties, model attribution and performance assessment (see Figure B.2). Although models are often considered as input schema, ESP-r can be used to add attributions (e.g. to raw CAD data) for export to other tools. The other outputs, building properties and performance assessment are described in this section.

B.9.1 Performance_assessment

Performance_assessment has been decomposed into three subtypes:

Principal_parameter_data is the raw timestep data of temperatures, flows and fluxes. These can be given for any node within ESP-r’s discretized system.

Derived_performance_data relates to commonly asked thermal design questions, such as energy consumption, plant sizing, comfort conditions, overheating risk etc.

Parameter_distribution_data Each of the temperatures, flows and fluxes can also be expressed in terms of their distribution (e.g. temperature profile through a construction, shortwave distribution within a zone).

Associated data: Zone_name holds the names of the zone(s) or "all" for the scope of the analysis reporting and Start_period and End_period define the period of analysis.

B.9.2 Principal_parameters

Principal_parameters (Figure B.20) are raw timestep temperatures, flows and fluxes. Subtypes are Zone_data, Surface_data, Intra-constructional_data and Flow_data. These can be expressed in several forms: Summary_data (i.e. maximum, minimum and mean values over the analysis period) or Timestep_data (data exported at the simulation timestep, or averaged over specified time-steps).

Zone_data consists of temperatures, fluxes (from heating, cooling, ventilation and infiltration) and relative humidity relating to the air in the zone.

Surface_data consists of the fluxes (convective, longwave and shortwave radiative, and conductive), together with surface temperature. It also contains a reference to the surface name and a surface index to indicate internal or external surface.

Associated data: Surface_index (position in list), Intra_constructional_data (data within the constructions) and Node (position within a construction).

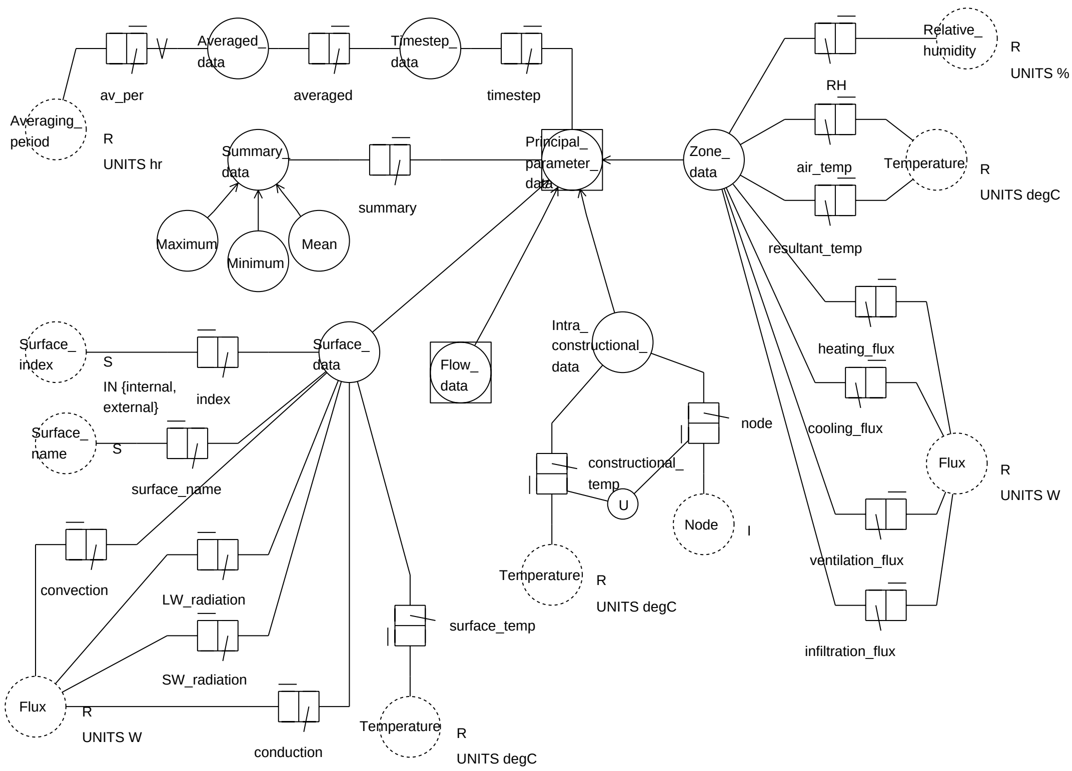


Figure B.20 Principal_parameter_data diagram (referenced by: performance_assessment).

B.10 Flow

B.10.1 Flow_network(s)

The **Flow_network** (Figure B.21) is the highest level entity on the network flow side of the problem description. Networks consist of nodes, components and connections. Both air and water flow networks are possible (with an easy extension to other fluids if necessary).

A **Flow_node** represents a bookkeepping point in a network at which pressures and temperatures are known or are calculated.

Flow_connection specifies the links between nodes, with one component linking each specified pair of nodes. Relative heights from the connection to the nodes are required.

Flow_component is a components used within the network (.e.g fan, duct, valve).

Associated data: Wind_speed_reduction_factor (to convert from the wind speed in the climate file to a building reference height).

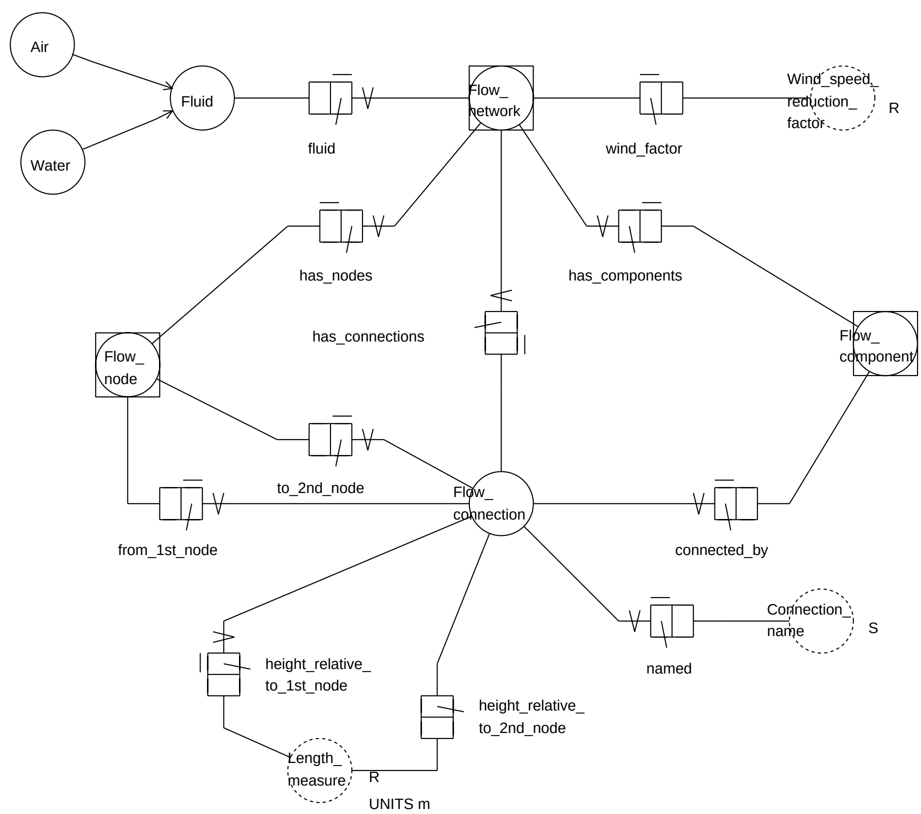


Figure B.21 Flow_network diagram (referenced by: ESP_r_problem).

B.10.2 Flow_node

A **Flow_node** (Figure B.22) represents a position in the network at which pressures and temperatures are known or are calculated. It can either be a boundary node or an internal node. Mass balance calculations are undertaken only at internal nodes.

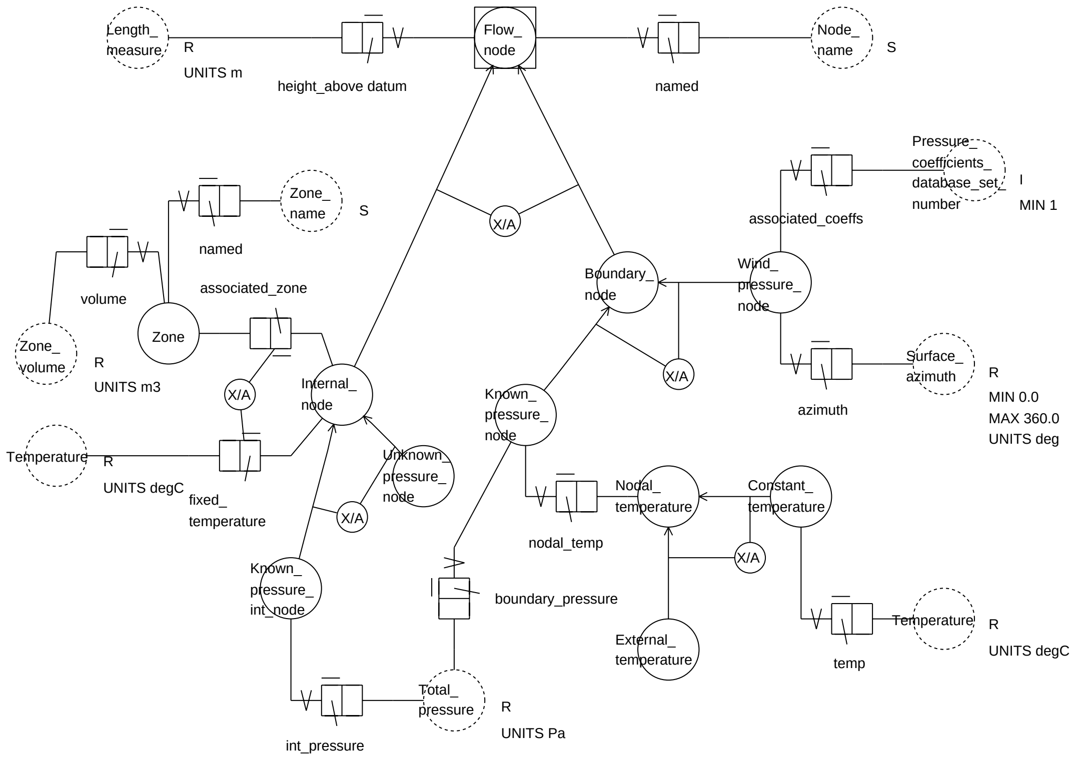


Figure B.22 Flow_node diagram (referenced by: flow_network).

A **Known_pressure_node** represents a position in the network at which pressures are known.

An **Internal_node** is usually an Unknown_pressure_node (and is calculated in the simulation) but may also be a Known_pressure_int_node. Its associated temperature can be either fixed, or calculated as part of the building simulation: in the latter case the associated zone is required.

A **Boundary_node** can either be a Wind_pressure_node (only for airflow networks but this constraint is not explicit on the diagram) or a node at which the pressure is known. If it is a wind pressure node, its temperature is automatically taken from the climate database.

Associated data: Pressure_coefficients_database_set_number (reference to the appropriate set in the pressure coefficients database), Surface_azimuth (degrees clockwise from north) and Zone_volume (for calculation of air change rates).

B.10.3 Flow_component

Flow_component(s) (Figure B.23) are components used within the network. Examples are flow resistance components such as fans and ducts. They can be referenced in one or more places in the network. Some of the components are suitable for a fluid type of air only (not shown in the diagrams). Each component has a Component_name.

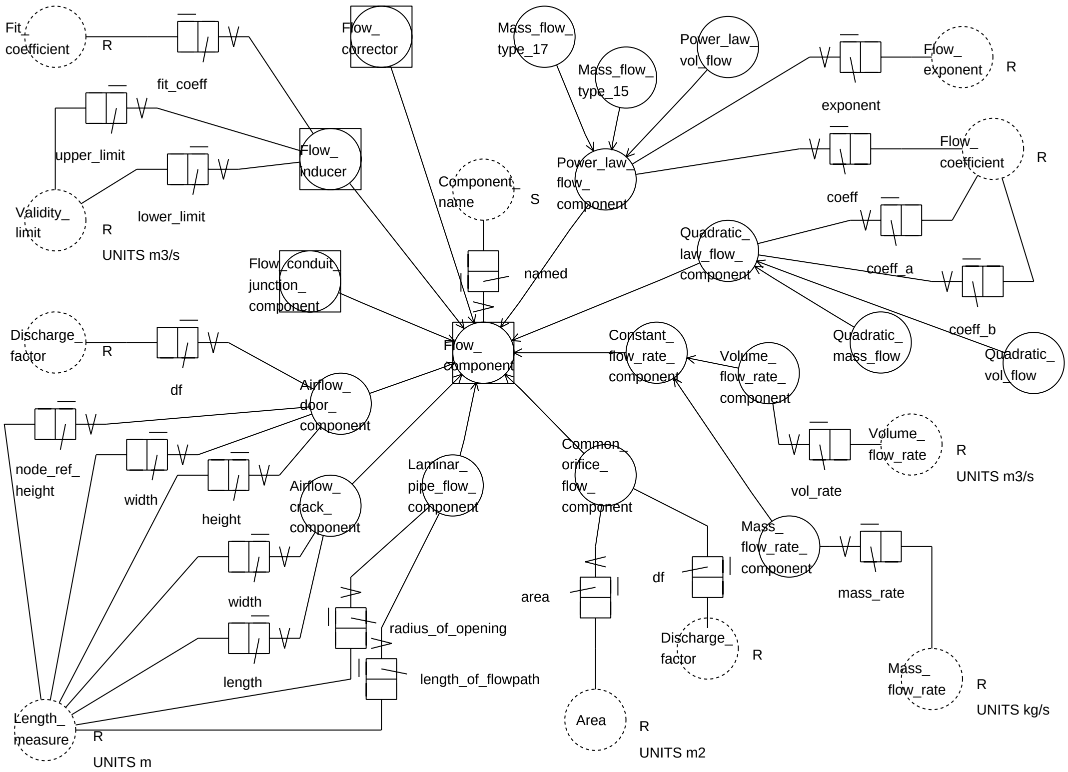


Figure B.23 Flow_component diagram (referenced by: flow_network).

Power_law_flow_component has three subtypes (Power_law_vol_flow, Mass_flow_type_15, Mass_flow_type_17) each with a different dimension for the flow coefficient. A Flow_coefficient and Flow_exponent must be specified.

Quadratic_law_flow_component has two subtypes (Quadratic_vol_flow, Quadratic_mass_flow) with different dimensions for the flow coefficients.

Volume_flow_rate_component is an ideal representation of a fan or pump requiring a volume flow rate.

Common_orifice_flow_component represents turbulent flow through relatively large openings.

Laminar_pipe_flow_component represents laminar flow along relatively long flow paths. Both the radius of the opening and length of flow-path are required.

Airflow_crack_component represents a crack. Both length and width are required.

Airflow_door_component is a Flow_component which supports bidirectional flow through a vertical opening. Both height and width of a door are required, together with the reference height of the neutral flow point and a discharge coefficient.

Flow_inducer represents a pump or fan. It requires a Fit_coefficient (coefficients a0 to a3 must be specified) as well as a Validity_limit within which the polynomial is applicable.

References

Augenbroe G. "Integrated building performance evaluation in the early design stages," *Building and Environment*, Vol. 27, no. 2, pp 149-161, 1992.

Hand J. and Strachan P., *ESP-r Data Model Decomposition*, ESRU Technical Paper 98/9, Energy Simulation Research Unit, University of Strathclyde, Glasgow, October 1998.

Appendix C

EXAMPLES OF SIMULATION USE WITHIN DESIGN

C Examples of simulation use within design

This appendix presents a sample of design projects to which simulation has contributed. Each example is briefly described in terms of the overall project, the nature of design questions which simulation was called upon to support, the metrics of the project, the approach taken and critical observations.

C.1 New buildings

Simulation has traditionally been associated with support for innovative projects and for projects whose performance is predicated on the interaction between unusual design features. In these contexts simulation's ability to focus on specific aspects of a design and the relationships between design elements is critical.

Potsdamer Platz Penthouse Study

Designs which include dramatic views and make use of direct solar gains must be balanced by consideration of controlling solar exposure in warm weather. This project involved the design of penthouse apartments in an exposed location which was subject to low angle solar penetration into highly glazed double height spaces. Design questions focused on the efficacy of various glazing systems and shading devices as well as on whether comfort could be maintained by the use of natural ventilation. The metrics of the project were hourly temperature patterns during design days, the relative frequency of thermal discomfort for each design variant over the summer season and the number of hours during which blind systems and mechanical ventilation were required to maintain comfort.

The approach taken was to focus on a single apartment design and compose a model at sufficient detail to monitor comfort, insolation patterns and air movement in each of the occupied spaces. The work proceeded in stages to identify performance patterns for design alternatives (e.g. different orientations, thermal mass and internal occupancy levels). The most promising variants were then used to assess different blind control and ventilation regimes. The last phase of the work fine-tuned the composition of the facade and the placement of mass and ventilation openings. The predictions indicated that critical control of the blind systems and placement of opaque facade elements were able to reduce the number of hours of discomfort, and that the addition of internal mass could delay the occurrence of peak temperatures.

Offices overlooking Edinburgh Castle

Occasionally the historical and urban context of a design complicates an otherwise straightforward project. In this project the proposed office accommodation had an unobstructed view of the castle in Edinburgh to the south and south-west. Local planning requirements dictated clear glass and minimal external solar protection. The design team wished to avoid the cost of full air-conditioning and proposed the use of a displacement ventilation system to control overheating. The design questions focused on the frequency of thermal discomfort vis-a-vis different glazing, blind systems and ventilation rates.

The approach was to model selected portions of the building at high resolution (e.g. each level was subdivided, raised floor plenums and return paths were explicitly represented). The assessments showed that encapsulated blinds were preferable to internal blinds and that blind control was critical to maintaining comfort. The assessments also indicated the climate conditions which would lead to localised discomfort and their frequency of occurrence. Subsequent investigations showed that a night purge ventilation regime would delay peak temperatures and that selective cooling of fresh air could shift this peak to after office hours.

Speculative design

The development of commercial property is often geared to attracting tenants of particular business types. One simulation project involved a "science park" of two-floor speculative offices for high-tech businesses and manufacturers. The developer was concerned that the provision of full air-conditioning would lead to unacceptable accommodation costs, but needed assurance about the risk of discomfort for prospective tenants in a naturally ventilated or forced ventilation scheme. The metric of the project was frequency of discomfort.

Because the proposed design included a mix of facade orientations and natural ventilation potentials, the approach taken was to model each lettable area in the building and to represent explicitly the facade and ventilation schemes. For each design and operational variant the assessments indicated the frequency of discomfort over the year for each lettable space. For those areas with poor ventilation potential, further studies identified the forced ventilation required and the level of internal gains which could be tolerated before air-conditioning was needed.

C.2 Refurbishment

The aging of buildings, changes in standards of accommodation or expectations of comfort and lighting are all driving forces for the refurbishment of existing building stock. Simulation has a role in supporting design decisions which prolong the life of a building and enhance the productivity of occupants.

Historical building in Dublin

Thermal simulation and daylighting assessments were used to support the refurbishment of a historically significant building to accommodate a government office, including a proposal to convert a courtyard into an atrium. The design team wished to know if the existing (historical) single glazed windows needed to be upgraded, whether the atrium might act as a buffer space, and the extent to which it might contribute to summer overheating or might reduce daylight availability in adjacent spaces.

The approach taken was to model the whole building and the atrium with sufficient geometric detail to represent the substantial building mass and fenestration details and to track temperature variations at critical points in the building. Insufficient information was available to support detailed flow assessments and so a range of scheduled flows was used. A separate daylight distribution analysis was also undertaken.

The assessments indicated that additional ventilation openings were required to balance internal heat gains in the summer and that the atrium would contribute to a 10-20% energy saving in the winter. The daylighting analysis indicated sufficient daylighting for somewhat over half the office accommodation and that the atrium would have only a limited influence on light switching decisions.

Sixties Office Block

Owners of 25-30 year old office blocks are confronted by a number of choices in refurbishment. Simulation can provide critical information. One such project focused on recladding and space planning options which would mitigate existing acoustic and thermal problems and would provide the building owner with options for attracting new tenants.

Simulation was used to support the design team in considering the trade-offs between different options early in the project. The remit was to respond quickly to "what if" questions, and the metrics and model details were required to evolve accordingly. The approach taken was to derive separate models to deal with macro issues (e.g. solar access and wind exposure) and detailed issues (fenestration, ventilation openings) and to ensure that the detailed models were designed to evolve quickly.

The work proceeded in phases which explored each of the proposed facade treatments and tested for sensitivity to orientation and occupancy types. A compromise facade treatment was found which could be used in the majority of contexts. The sensitivities of the building to restrictions in cross ventilation and to the isolation of structural mass were also determined. In most cases, responses to "what if" questions were delivered within 24 hours.

C.3 Change of use

Another issue for the design professions is the conversion of existing building stock from one use to another. The proposed use may have different thermal and visual requirements from that for which the building was initially designed. In extreme cases, e.g. a change from warehouse to office accommodation, the conversion may require significant alterations to environmental systems. Simulation is able to provide indications of alternative fabric, fenestration and environmental systems which are appropriate for the new use.

Warehouse Conversion Study

A typical example of this type of project involved the conversion of a three storey warehouse into an architectural office. This radical change in use posed design questions on all aspects of the inherent performance of the building shell and its response to various operational and control regimes. Design questions related to the selection of facade components and to the sensitivity of performance to alternative internal layouts such as the introduction of an internal lightwell, vertical circulation space and the efficacy of natural ventilation for summer cooling. The metrics of the project were cost, comfort, daylighting levels and environmental system simplicity.

The approach taken was to focus on the upper two floors of the warehouse in detail and to approximate the balance of the building. The model was designed to support the simultaneous evaluation of different facade and glazing variants and operational regimes so that sensitivities could be identified. A daylighting assessment was also undertaken. The project deliverables were a set of performance predictions and the rules by which the client could judge the contribution of various design and operational alternatives.

C.4 Educational buildings and community centres

Owners of building stock, such as local government bodies, are confronted by design and operational questions for existing facilities and new facilities which must balance capital and running costs. Their decision making process can benefit from simulation-derived information.

High School Study

School complexes are often required to adapt to changes in demographics and curricula. In one such project a proposal for a new classroom wing for business studies appeared to require alterations to an existing central boiler facility for the campus. Simulation was used to evaluate whether it was possible to proceed without costly changes to the central plant. The options investigated were the management of existing demands (e.g. plant control and delivery regimes, and energy conservation) to recover capacity for use in the new wing, to design the new wing to be autonomous and to evaluate supply-side options such as the addition of a top-up boiler.

The metrics of the project were plant capacity and running costs, and comfort within existing facility and the proposed extension. The approach taken was to begin with an investigation of one wing of the existing building and to test control and delivery alternatives such as zoning classrooms and sequencing heat delivery. Energy conservation options were then investigated.

The second phase of the work was to investigate the proposed extension with particular attention to the limitation of peak heating demands. This involved assessing construction details, the use of internal gains to offset heating demands and the use of heat exchangers to reduce fresh air heating loads. Design options were also evaluated for possible adverse effects on summer comfort.

The assessments from the initial phase showed that zoning of classrooms and sequential delivery of heating reduced capacity but required an earlier system start. Energy conservation options and related demand reductions were communicated to the design team for consideration against energy conservation costs in the new wing. The second phase assessments indicated that heat exchangers did not provide a reasonable return on investment but that early light switch-on could be used to displace critical central plant capacity. The proposed lightweight construction was found to have insufficient mass to absorb transient internal loads and it was recommended that some masonry elements be introduced.

Hospital Dayroom

This project involved a hospital dayroom which had been designed as a conservatory but presented a risk of thermal stress to patients if temperatures changed quickly. Observations showed that some patients were not capable of judging when to move away from overheated conditions or of asking for corrective actions. Design questions focused on the efficacy of motorised vents or fans for ventilation cooling and blind control for limiting insolation into the dayroom. The metrics of the project were thus comfort and response to changing environmental conditions.

The approach taken was to model the dayroom and the adjacent ward and to begin by assessing solar access to the dayroom and then to test the efficacy of natural and forced ventilation and blind control. It emerged from the assessments that glare was also a consideration. Assessments showed that a combination of blinds and motorised vents was generally adequate and defined the conditions under which an extract fan would be necessary.

Community Resource Centre

The design of small scale community facilities often proceeds with standard design details and environmental systems. However, these sometimes require adaptation to meet the requirements of disabled or elderly occupants and novel approaches sometimes require careful consideration. In one such centre, the services consultant proposed using a heated ceiling system to eliminate accidental contact with radiators and to free wall space. Simulation was used to investigate whether the design would provide adequate control and would react to variations in climatic patterns and occupancy loading.

The approach taken was to model three rooms in the centre, of which two were subject to variations in occupancy and one was sensitive to boundary conditions. Care was taken to agree on sets of climate data which included both typical and extreme conditions. A solar access study was undertaken to identify critical portions of the building. The ceiling heating system was explicitly represented so that its response could be tested and possible radiant discomfort checked.

The assessments indicated that comfort could be maintained without elevating the mean radiant temperature of the rooms for extended periods. They also indicated that back-losses from the ceiling heating made comfort control of the upper floor difficult unless additional backing insulation was provided.

C.5 Confirming the efficacy of a building prototype

Although practitioners do not often consider simulation to be economical for use in dwelling design, the operational and construction savings on multiple units can help offset design costs in the case of prototype housing units.

Buffer House Study

One such study involved the design of an innovative "buffer house" for use by a Scottish local authority in areas of high wind and rain exposure. The design concept centred on the design of a small space and energy efficient house ($54\,m^2$) with an attached $27\,m^2$ highly glazed buffer space for use in drying clothing, and providing a sheltered work space and children’s play area. The design questions centred on the range of environmental conditions which might be expected in the buffer space, the extent to which it might act as a ventilation preheat for the house, optimal levels of insulation in the dwelling and the impact on solar gains of incorporating deep window reveals in the facade.

The metrics of the project were temperatures in occupied spaces and in the buffer space, heating capacity and demands, and daylight distributions. The approach taken was to model the whole house with additional detail in the facade and buffer space to allow the distribution of solar radiation and air movement to be assessed. One variant of the model was composed to test sensitivity to orientation, otherwise each design option was applied sequentially. The predictions indicated that the buffer space would be usable throughout the year, would preheat fresh air and that the annual heating costs for the dwelling would be considerably less than the norm.

C.6 Urban scale designs

Development at the urban scale can benefit from a balance of information on macro scale performance as well as testing the performance implications of a policy decision on individual dwellings or offices.

Microclimate Studies

One aspect in the planning of housing developments is an assessment of site conditions. Simulation has been used in several microclimatic studies to investigate issues of solar access and wind exposure, and to provide design guidance for the dwellings to be built on the site.

The metrics of these projects were hours of direct solar access, indices of wind exposure, patterns of sunlight across the site and variations in energy demands. The approach taken in these studies was to evolve both urban scale and dwelling models and to apply the visual and thermal information gained in each to the needs of the project. In both cases initial assessments began with reviews of climatic patterns and the topology of the site. In one case the site was on the north slope of a hill. To assess solar access and wind exposure one street of typical dwelling types was modelled and the explicit site context extended to approximately one kilometre to the south, east and west of the street. Information was delivered in the form of a sequence of images of the site and of specific dwellings at different times and seasons. Frequency bins of energy demands for different locations on the site and with different standards of construction were also provided.

Solar Quarter

One urban planning scheme focused on the redevelopment of a mixed use neighbourhood on an island in the town of Regensburg. The goal of the project was to evolve a neighbourhood with considerable energy autonomy and a local co-generation facility was one part of the scheme. The design team was concerned to balance high density with solar access and energy conservation. Design questions included the massing and placement of buildings, and the pattern of power demand on the island. The team also wished to publish design guidelines for apartments and business accommodation to be built under the scheme. The metrics of the project were insolation patterns, detailed and aggregate energy demands, and co-generation heat-to-power ratios and their sensitivity to building composition options.

The approach taken was to use visual assessments of massing studies to address macro issues, a series of detailed models to identify sensitivities to design details and the scaling and aggregation of detailed assessments to find urban scale performance indices. One of the challenges of the study was ensuring that a range of design variants and usage patterns was included in the underlying studies and that the design team was supplied with an appropriate balance of information from the detailed and macro studies.

C.7 Testing design features

Consulting projects often identify specific facets of a design which are not amenable to conventional assessment. In such cases simulation has been used for focused assessments.

Reception area in an atrium

One simulation study focused on conditions in a reception area at the base of an atrium in a proposed office building. The atrium extended through three floors of the building. The design team were concerned that cold downdrafts might occur in the winter, overheating might be experienced in the summer and that environmental conditions might not stabilise early enough on Monday morning after a weekend operating regime. The design team proposed the use of radiant panels mounted on the perimeter of the atrium at the first floor level.

The metrics of the project were air velocity and temperature profiles in the atrium and comfort in the reception area. The approach taken was to model the atrium in some detail, with less detail in adjacent spaces which defined boundary conditions. The assessment showed that the radiant panels balanced the cold surfaces in the roof of the atrium and ensured comfort during office hours. The flow predictions showed that cold air adjacent to the roof rarely reached the occupied space and that summer overheating was unlikely.

Evaluating an "Energy Wall"

Design teams often use a central design metaphor to influence the overall plan of a building. In one study this core idea was an "energy wall" on the facade of a library which would provide light to circulation spaces and isolate the occupied spaces from variations in outside temperatures without resorting to the expense of double glazing. In order to confirm the efficacy of this idea, simulation was used to evaluate the performance of the facade and to predict the conditions within the adjacent spaces in terms of energy demands, comfort, daylighting and glare.

The approach taken was to model the energy wall and the adjacent spaces at a moderate level of detail. The model was also used as the basis for visual assessments. The importance of air flows to the operation of the energy wall dictated the inclusion of network flows. The assessments showed that lighting energy savings were possible but that visual glare discomfort was probable under specific climatic conditions and that blind control was critical. Assessments also showed that the energy wall would act as a buffer space in moderate conditions and that controlled venting of the energy wall would limit summer overheating.

Evaluating integrated photovoltaic facades

Component design is often treated in isolation (analytically) or assessed under laboratory conditions which are not indicative of realistic operating regimes. One example of this is the application of photovoltaic modules to building facades. Module performance is influenced by their integration in the facade but design guidelines do not often take such information into account.

Simulation is able to assess the electrical performance of modules with realistic boundary conditions and patterns of radiation and, importantly, is able to assess how the thermal energy associated with photovoltaic modules can be applied to the demand profiles of a building. The approach taken in one such project was to compose and calibrate a model of an empirical test module, and then apply the design feature to a range of buildings. The result of this project was a set of rules for hybrid electrical and thermal photovoltaic applications.

C.9 Identifying design faults and remedial work

The application of standard design details can lead to unintended consequences which simulation is able to identify. Building faults such as overheating, condensation or poor ventilation are issues which have been addressed by simulation.

Balancing natural ventilation flows

An example is the assumption that the specification of building code compliant ventilation openings is a conservative design approach. In one project a reception area in a community health facility was under an atrium and nominal openings were included in the atrium and in the facade of the building.

When represented as a model the predicted performance indicated a risk of overheating in the summer and insufficient fresh air during calm cold winter conditions. The explanation was that the window vent design and atrium had been considered separately and the roof level vents had insufficient inlet area and so in calm conditions stack effects were unable to deliver adequate ventilation until such time as overheating induced additional buoyancy driven flows.

Condensation study

One investigation involved a community sports centre with an indoor running track which experienced condensation accumulation on the track surface. The running track was housed in an unheated tented structure and condensation formed on the inside fabric skin of the structure and fell on to the running track under particular climatic conditions. Simulation was used to explore how to reduce the risk of condensation. Options included insulating the structure, installing underfloor or radiant heating systems, lowering the humidity in the space by enhancing ventilation or mechanically removing moisture from the air, or some combination of these approaches.

The approach taken was to model the whole structure and to design a model to capture surface and air temperatures and moisture flows at critical locations. A series of assessments, each focusing on one design or control variant was run and the overall performance evaluated against measured conditions. Assessments found that one dehumidifier reduced 25 days of condensation risk to four days and two dehumidifiers resulted in three hours of risk. The combination of radiant panels and dehumidifier removed condensation risk and improved comfort for runners.

ESP-r EXEMPLAR MODELS

D ESP-r Exemplar models

This appendix presents an overview of the conventions and structure of exemplars. Access to well documented and well structured example problems (exemplars) is supported by documentation and attribution extensions of the ESP-r data model and enhancements to the problem configuration which allow models to be relocated easily (see Section 5.1.3). The tight binding of the interface to the data model also allows more of the context of each entity to be displayed.

D.1 Conventions

In the current work an exemplar is a model which has been purposefully structured, annotated and documented to support ease of understanding and which demonstrates clarity of purpose. It is important that exemplars are syntactically correct, are well abstracted representations of the design and are commensurate with the metrics and methods of the project. This definition was arrived at by taking models from consulting and research projects and identifying specific barriers to their being understood. Chapter 3 identified these barriers, Chapter 4 suggested mechanisms to address them and Chapter 5 put in place the necessary functionality. Examples of the mechanisms follow.

The structure of models

ESP-r uses a partitioned file store, e.g. schedules are held separately from zone composition, and as discussed in Section 5.1.3, separating data by type yields a number of benefits. However, no particular rules of file store distribution were enforced and those who were unwilling to impose their own structure could find the project folder included upwards of 100 files. In extreme cases (see the upper portion of Figure 4.2) the resulting clutter imposed a management burden. To address this, the Project Manager took on the task of distributing data within a standard hierarchy of folders such as that shown in Figure D.1

```
simple/
   cfg/
      bld_simple.cfg     bld_simple.cnn     bld_simple.log
      bld_simple_fzy.cfg bld_simple_fzy.log bld_simple_shd.cfg
      bld_simple_shd.log bld_simple_ufh.cfg bld_simple_ufh.log
   ctl/
      bld_simple.ctl     bld_simple_fzy.ctl bld_simple_ufh.ctl
   images/
   zones/
      reception.con reception.geo reception.obs reception.opr
      reception.tmc
```

Figure D.1 Distributed file store example.

Projects which follow this pattern place system configuration files in the *cfg* folder (several configuration variants are included in the example), control definitions are located in a *ctl* folder and zone files are placed in a *zones* folder. As a project increases in complexity new information is distributed accordingly. Individual data store files also follow a naming scheme—system files share a root name and files related to each zone share the root name of the zone. Hardly a radical idea, and yet such steps do have a noticeable effect on the resources required to manage projects and to carry out simulation tasks.

The concept of a distributed and partitioned data store can also be expressed at the corporate level. For example, consider the training exemplars associated with the standard distribution of the system shown in Figure D.2. Here exemplars are arranged by topic (e.g. the plant folder contains 10 variants) and by project type (e.g. office building, low energy house). The multiple project configurations in Figure D.1 represent incremental variants of a simple building used to demonstrate particular aspects of simulation (e.g. several exemplars share a common directory structure). There are also other structures which can be used to archive projects.

```
esp-r/
   training/
      Gridding/  basic/     exemplars  mould/     office/    pid/
      pv_facade/ simple/
      cfd/
         IEA_A20/     RoomVent98/  displ_vent/  rad_htg/     readme
         template.bmp template.dfd
      cg_ctl/
         coupling/    daylit_coef/ static/
      chp/
         sport_cen/ unit/
      constr/
         adapt/  tp_sub/
      house/
         sun_space/ svph/
      plant/
         ac_pp/          ahu/            blyth/          coil_pp/
         conv_ac_sys/    mixed_ac_sys/   solar/          vent_detailed/
         vent_simple/    wch/
      students/
         cool_coil/ cool_twr/  gas_he/    liquid_he/
```

Figure D.2 Distributed file store example.

The Project Manger provides browsing facilities (see Figure 5.1.7) via the mechanism of an *exemplars* file (extracts are listed in Figure D.3). This list includes the provision for commentary for each group of projects and each "*item" includes a selection label, an aide memoire, the name of the system configuration file (the primary unit of identification and exchange) and the root folder of the project. The latter allows the Project Manager to take a copy of the project.

```
*EXEMPLARS
*help Some typical designs to provide a basic introduction
*help to ESP-r problem definition and illustrate realistic
*help applications.
*group  basic training
*help A set of models demonstrating ESP-r’s functionality.
*item
*name 1 zone with convective heating and ideal control
*aide L-shaped zone with convective heating system and ideal control.
*cfg  /usr/esru/esp-r/training/simple/cfg/bld_simple.cfg
*root /usr/esru/esp-r/training/simple
*item
*name as ‘a’ but with underfloor heating
*aide L-shaped zone with underfloor heating system and ideal control.
*cfg  /usr/esru/esp-r/training/simple/cfg/bld_simple_ufh.cfg
*root /usr/esru/esp-r/training/simple
 . . .
*item
*label  . . . .
*group  more realistic basic training
*help A set of models demonstrating ESP-r’s basic modelling features.
*item
*name house with sun space, winter operation
*aide House with a sun space and winter-typical leakage distribution.
*cfg  /usr/esru/esp-r/training/house/sun_space/cfg/house_win.cfg
*root /usr/esru/esp-r/training/house/sun_space
 . . .
*item
*label  . . . .
*group  plant modelling
 . . .
```

Figure D.3 Fragments from exemplars list.

Naming schemes

Many entities in an ESP-r model are named as they are created. As discussed in Section 4.4, it is difficult to name something without having come to some understanding of its nature and the subsequent use of logical names adds clarity to user interactions. Consider the surface attribution and surface selection list shown in Figure D.4. To the tool designer these are object expressions of the data model. To the practitioner they ensure clarity of selection and understanding. The metaphor is one that works and other design decision support tools would do well to adopt such conventions.

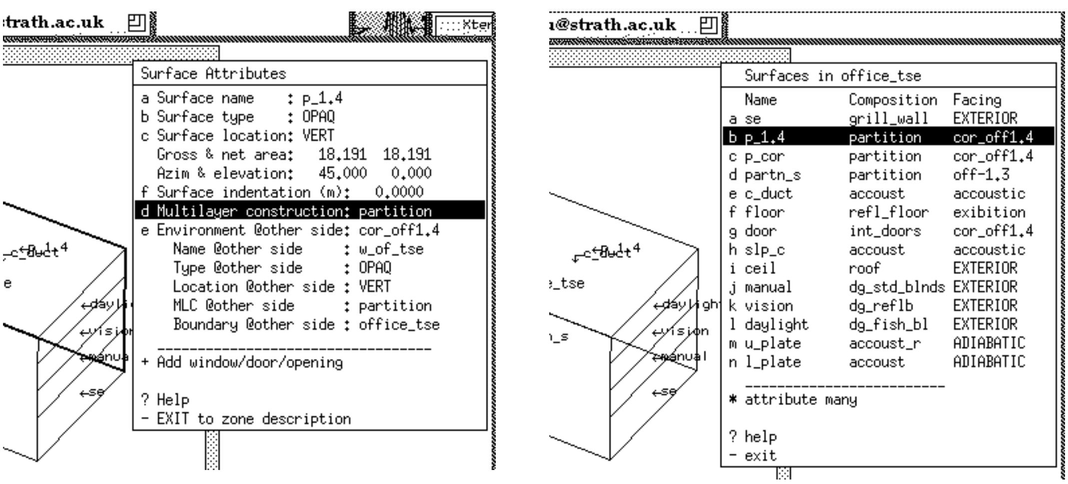


Figure D.4 Logical names in aid of surface attribution (left) and surface selection (right).

Project notes and documentation

The case studies provided evidence of project information which was held on paper rather than as part of the definition of the model. For example, the temporal use of a building is encoded as a list of numbers (see Figure D.5), which are more than sufficient for computational demands, but lack clarity for the user. Here the approach taken was to extend the operational data model to include a descriptive phrase such as "lights 7h-22h, 6 people @ meals" to record the intent of the schedule.

```
[fragment from composition report]
 . . .
 Description : day room operation - lights 7h-22h, 6 people @ meals
 No. of Weekday casual gains =  6
 Gain Type Start Finish Sensible  Latent    Radiant   Convec
           Hour  Hour   Magn. (W) Magn. (W) Frac      Frac
   1 Occupt    0 24     60.0     30.0       0.60       0.40
   2 Occupt    8 10    600.0    600.0       0.60       0.40
   3 Occupt   12 14    600.0    300.0       0.60       0.40
   4 Occupt   17 20    600.0    300.0       0.60       0.40
   5 Lights    7 22    110.0      0.0       0.50       0.50
   6 Equipt    7 22    100.0      0.0       0.50       0.50
 . . .
```

Figure D.5 Documentation in support of numerical data.

It is critical that the metrics and assumptions of the project are held with the model. In the case of training exemplars this documentation might contain highlights of the model and details for the student to explore. The listing in Figure D.6 is an extract from such a project document.

Ideally such documents record the initial intent of the project and early assumptions at the registration phase and are updated as the project progresses. Information such as the spatial distribution data in the middle of the list would be critical to a future replication study as well as supporting current reporting requirements.

Project images

Project documentation acts both as an archive of reference material and a mechanism for supporting the understanding of models. Section 4.2 discusses the importance of the rapid assimilation of models. This process is dependent on visual clues as well as written documentation and model browsing facilities. Figure 5.5 shows examples of montages associated with models. Often pictorial evidence (e.g. site photos, facade renderings) is available early in a project. The Project Manger registration facility allows such information to be associated with the model. Montages of model details and performance patterns have also been associated with models to assist managers in scanning archives for relevant examples for new projects or for briefing potential clients.

When images are registered with a project they can be tagged by topic (e.g. zone composition, networks, control, computational fluid dynamics, performance) to be invoked only in association with that topic. Figure D.7 shows an *Active definitions* display which includes an image available symbol in the zones box.

```
[fragments from a project log]
 . . .
The project is based on Solar House project General Teaching
Hospital Thessaloniki (Meletitiki - Alexandros Tombazis & Assoc.
Architects Ltd). The Teaching Hospital design has been altered to
take into account the distribution of spaces in a typical 300 bed
hospital extracted from DHSS Low energy hospital study [BDP 1985].

The base case is equivalent to best practice hospital
design with conventional energy conservation measures
(i.e. moderate insulation levels, double glazing, efficient
lighting and environmental systems).
 . . .

The distribution of spaces is based on the pattern from an adult
acute module of a Nucleus hospital as follows:

ground floor:            %      m^2
 6 bed ward             8.0%   49.   (zone 15)
 single beds (4)        6.5%   40.2  (zone 10)
 reception              4.0%   24.5  (zone 11)
 staff/office/records   8.0%   49.   (zone 13)
 passages+waiting       9.0%   54.56 (zone 9)
 theatres               5.0%   30.6  (zone 12)
 catering               4.0%   24.5  (zone 16)
 support                5.5%   34.0  (zone 14)

first floor:
 6 bed wards (2)       16.0%   98.0  (zones 7 & 1)
 4 bed ward  (1)        6.5%   40.2  (zone 2)
 single beds (4)        6.5%   40.2  (zone 4)
 dining/dayrooms        4.0%   24.5  (zone 3)
 passages               8.8%   54.5  (zone 6)
 baths                  4.0%   24.5  (zone 5)
 staff/office/records   4.0%   24.5  (zone 8)

The model represents roughly 10% of a 300 bed hospital.

Temperature control is as follows:
a) wards, baths 20 C at all hours, no cooling
b) corridors 18 C at all hours
c) theatres, examination, reception, offices 20 C 7h00-18h00 then 18C
   theatres are cooled if > 24C
 . . .
```

Figure D.6 Project document.

D.2 Exemplars in training

Although the possibilities opened by linking exemplars and training materials are only beginning to be explored, the evidence thus far indicates that considerable potential exists. A list of the exemplars current at the time of this writing is included in Table D.1.

The exemplar mechanism allows the novice to browse a range of models in rapid succession in a way that de-emphasises the interface. It supports instructors in directing the focus of attention (e.g. it allows the simulation equivalent of "turn to Section 4.1 on the effects of mass on overheating"). Instructors who are so motivated can compose a sequence of bespoke models which match their approaches to training.

Beyond the remit of exemplars at this time are mechanisms to clarify for the true novice the semantics of what is represented on the workstation display. Without foundation skills the interface is often no

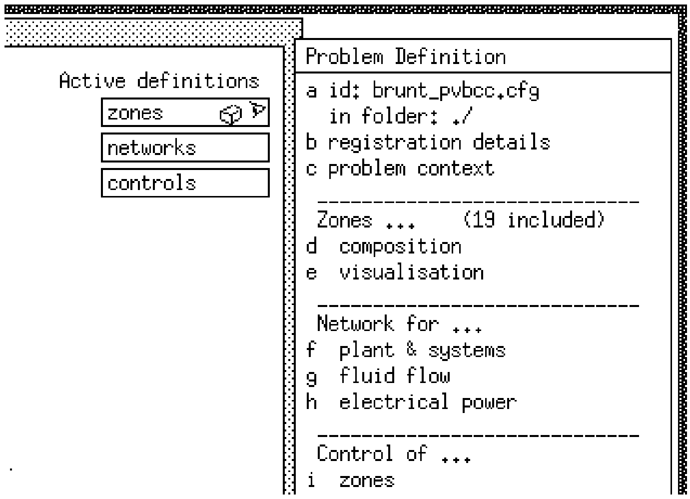


Figure D.7 Active definitions and images.

more than lines and words. This said, the interaction potential of hypertext may provide opportunities for the instructor to help the novice to use exemplars as a mechanism for acquiring foundation skills.

Table D.1 Current training exemplars.

Topic	**Exemplar description**
basic training	L-shaped zone with convective heating system and ideal control. L-shaped zone with underfloor heating system and ideal control. L-shaped zone, external shading, convective heating and ideal control. 3 zones, convective heating and ideal control. 3 zones, standard model with external shading, explicit internal longwave exchange. 3 zones, standard model with scheduled infiltration added. 3 zones, standard model with scheduled infiltration and controlled ventilation. 3 zones, standard model with controlled blinds. 3 zones, standard model with control set to match adjacent zone temperature.
more realistic basic training	House with a sun space and winter-typical leakage distribution. House with a sun space and summer-typical leakage distribution. House with solar ventilation pre-heat from a conservatory. Office with natural ventilation, (configured for summer operation). Office with photocells for luminaire control (assumes static illiminance). Office with photocells for luminaire control (uses direct coupling with visual simulation). Office with photocells for luminaire control (uses daylight coefficients). Building with convective heating system and PID control (both dynamic and steady-state PID used). Zone with convective heating system and fuzzy logic control.

Topic	Exemplar description
plant modelling	Mechanical ventilation system (simple system with no building). Mechanical ventilation system (detailed system with no building). Air-conditioning unit (including humidity control) serving one zone. Wet central heating system serving one zone. Air conditioned building with humidity control. Ventilation heat recovery. 4 zones, each serviced by an alternative active solar arrangement. Fan duct system with zone heater (modelled using "primitive parts"). 4 row cooling coil (modelled using "primitive parts").
high resolution constructions modelling	Time dependent thermo-physical property substitution. Transparent insulation with variable thermophysical properties. Adaptive 1-D gridding. Adaptive 3-D gridding. 3-D ground conduction. Model supporting moisture flow analysis. Mould growth detection. Mould infested house analysis.
CFD modelling	Analysis of a radiant heating system (CFD not active). Analysis of a radiant heating system (CFD active). CFD analysis of a displacement ventilation system. Comparison with IEA Annex 20 isothermal test case. Comparison with IEA Annex 20 non-isothermal test case.
combined heat and power	Passive combined heat and power using photovoltaic facades. Engine-based combined heat and power. Sports centre with combined heat and power plant.

References

BDP Energy and Environment, *Planning Report: UK Department of Energy Passive Solar Design Studies, Non Domestic Buildings, Second Phase: Atria and Conservatories*, Building Design Partnership, Manchester, UK, June 1989.

E COMBINE project overview

This appendix presents an overview of the COMBINE project. The following paper appeared in the proceedings of Building Simulation ’95 of August 1995 in Madison Wisconsin.

The Development of an Intelligent, Integrated Building Design System Within the European COMBINE Project

J A Clarke*, J W Hand*, P A Strachan*, D F Mac Randal#

There are two main issues to be resolved in order that design tools can be used in cooperative mode, each communicating with the other. Firstly, there is a need to put in place a consistent product model of a building and its systems from which disparate design tools can obtain their inputs and return their outputs. Secondly, there is the requirement to manage the transactions between users and design tools. These issues were addressed within the European COMBINE project. This paper is concerned with the latter issue. It describes the basis and operation of an intelligent, integrated building design system, or IBDS, which is able to coordinate designer-to-designer, designer-to-application and application-to-application transactions, against rules which describe the purpose of a given design session. The IBDS is able to address ‘shallow’ control, where the design tools are sequenced, and ‘deep’ control, where knowledge is introduced in relation to design purpose so that design tool use is constrained within a given design session.

INTRODUCTION

To bring real benefit, building performance modelling must be integrated within the design process. Traditionally, as summarised at the left of Figure 1, the use of design tools has followed a *tool-box* approach in which the designer is expected to recognise a particular task, locate a suitable program, run it and translate its outputs to appropriate changes to the design hypothesis.

Clearly this is an inadequate approach in that the tools are decoupled from the design process and require the designer to be knowledgeable about each tool’s capabilities, control syntax and semantics.

* ESRU, Energy Systems Division, Faculty of Engineering, University of Strathclyde, Glasgow G1 1XJ. email: esru@strath.ac.uk, phone: +44 141 552 4400 X3024, fax: +44 141 552 8513.

Informatics Department, Rutherford Appleton Laboratory, Chilton, Oxon. 0X11 0QX. email: damian@inf.rl.ac.uk, phone: +44 1235 445403, fax: +44 123541 44589.

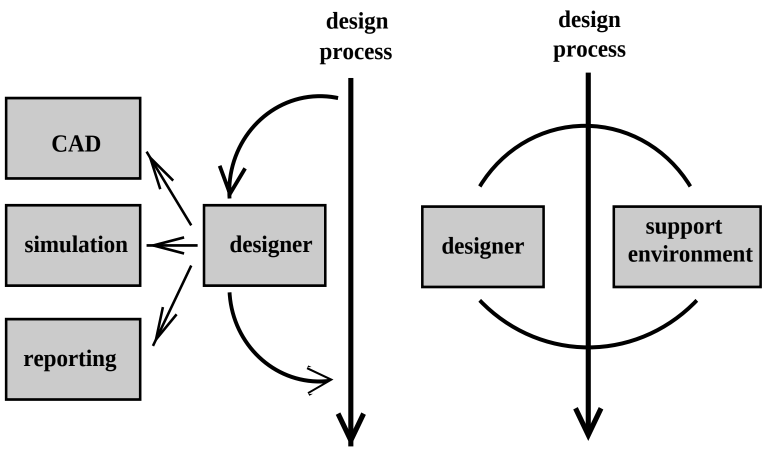


Figure 1: Tool-box and integrated approachs to design. (After MacCallum 1993)

An alternative approach is summarised at the right of Figure 1. Here the computer resource is (somehow) integrated within the design process. In such a *Computer-Supported Design Environment* (CSDE), the designer evolves the design hypothesis in such a way that the tools are able to automatically access the data describing the design and give feedback on all aspects of performance and cost in terms meaningful to the designer.

The attainment of such a CSDE is a non-trivial task involving the development of *integrated product models* and *intelligent interfaces*.

In the former case, the complexity stems from the temporal dimension of the design process, i.e. the evolution of the describing data against an uncertain information context and the different professional viewpoints and vocabularies. Within the European Commission's COMBINE project (Augenbroe 1992), the objective was to evolve an integrated data model (IDM) which could satisfy the needs of a representative set of design tools (for energy analysis, CAD, lighting, regulations compliance, layout planning and the like). The IDM (or strictly speaking the Data Exchange System, or DES, which is an implementation of the conceptual IDM) is then able to receive from, store and deliver data to these design tools in a manner which ensures that these data are accompanied by their related semantics: in the COMBINE project the EXPRESS language (Spiby 1991) is used to achieve this end. A key issue is the structure of the data model to ensure efficient exchange and allow future extension as additional or more powerful design tools are added. The decision to base this data model on the object oriented paradigm and to contain it within an object oriented database was seen as the way to achieve these goals.

The development of an intelligent interface, the subject of this paper, is non-trivial because of the complexity of the transactions which require to be managed in terms of:

- supporting design concurrency (designer to designer and designer(s) to application(s) inter-communication)

- preserving audit trail (who did what, when and why)
- supporting a constructive user dialogue (style of interaction, feedback and tutoring)
- evolving the product model (incremental problem definition and intelligent defaulting)
- and handling application semantics (application to product model and application to application).

One COMBINE task explored the form of an intelligent, integrated building design system, the IBDS, which could handle these issues. This was done via a rapid prototyping approach by which different scenarios for design tool transaction management and data exchange were explored. The prototyping environment was the IFe system (Clarke and Mac Randal 1993), the modules of which are:

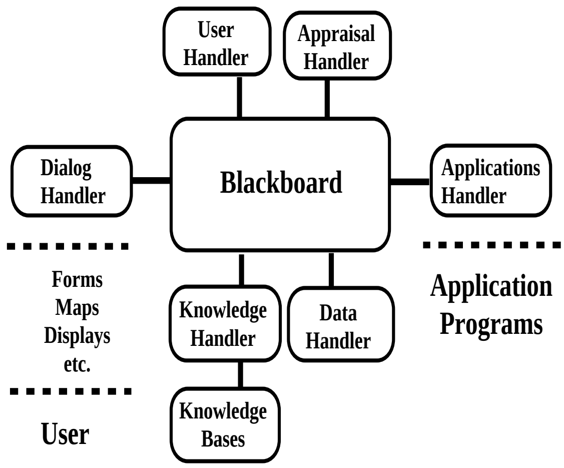


Figure 2: The IFe System.

- A Blackboard to serve as a communication centre for its various clients. By this means concurrency can be supported and traceability achieved through the collection, organisation and storage of the session chronicle.
- An Application Handler to control the various design tools, pass them their data and receive their returns.
- A Knowledge Handler to control design tool access to the product model and the communication with the designer (verification of entries, supplemental inferencing and feedback/guidance).
- A Dialogue Handler to converse with the user by means of acceptable concepts which relate to the different user types and levels of expertise.
- A User Handler to track the user’s progress and ensure the system responds in an appropriate manner.
- An Appraisal Handler to hold the design tool control syntax against standard performance assessment methodologies.

- A Data Handler to extract an application's data from the Blackboard and organise these data in the required format.

The IBDS is therefore based on a Blackboard/Knowledge Handler architecture to effect purpose-specific design tool control. It incorporates several real design tools (DTs), which can be configured to support real design sessions.

IBDS DESIGN TOOLS

The aim of the IBDS prototype was to provide a mix of design tool functions (DTFs) by which a number of realistic design sessions could be accommodated, each to a realistic level of complexity. While some DTFs require substantial interactions with the user (e.g. a CAD program), others may be purely computational (e.g. a regulations compliance program) so that the user need not be made aware of their existence. The DTs known to the IBDS at the present time are as follows.

Architectural CAD

Two Architectural CAD DTs are supported: AutoCAD Release 12 (Autodesk 1989) running in native mode with a constrained set of commands and drawing entities consistent with the DES and MicroStation (Conforti 1994) configured as an on-line interface to the DES.

Geometrical Attribution

The attribution of the problem geometry generated by AutoCAD is accomplished via the "Project Manager" module of ESP-r (Hand 1994), hereinafter called ATTRIBUTE. Attribution is in terms of construction, occupancy and control.

U-Value Compliance

The regulations compliance of a design is assessed by BRC (Rode 1993) which relates to several European national building regulations.

Thermal Energy

ESP-r's "Simulator" (Clarke 1985) module is included to enable performance evaluations such as summer overheating extent, winter heating plant sizing and heating energy requirement estimation.

Daylight/ Visual Impact

RADIANCE (Ward 1994) is included to enable the quantification of a zone's illumination levels and the production of visual impact information for the overall building. It accepts problem descriptions as generated by the ATTRIBUTE DT.

PROCESS MODELLING

In dealing with the design process at the level at which COMBINE operates the IBDS must support the flow of data/information between work-steps (or DTFs) and event-handling in terms of starting and stopping the design tools. The mechanism adopted to handle these issues within the IBDS is as follows.

The required process model is captured in the form of a Petri-Net (Javor 1993) and then this is transformed into a file of Prolog facts. This gives the basis of the process as a formal description. This file (hereinafter termed the PNF for Petri-Net File) is then dynamically read into the IBDS' Application Knowledge Handler (AKH) where it is used by a "process support inference engine" to animate the process. By modifying this process knowledge base (as held within the AKH), it is then possible to control the rigidity of the system, its handling of parallelism, etc.

In the current IBDS, three process models are available (each with potentially many instances) corresponding to:

Case 1: where DT invocation is not sequenced nor functionally constrained so that the designer is able to invoke the DTs in any order and activate their internal functions as required. That is the PNF is used only for DT access control.

Case 2: where the DTs are sequenced but not functionally constrained so that DT selection is prescribed while function invocation is not. That is the PNF controls DT ordering but DT use is opportunistic; concurrency is allowed.

Case 3: where the DTs are both sequenced and functionally constrained so that the system, not the designer, controls the order of DT selection and the invocation of the DTF. (But note that it is the DTFs that are being automated, not the design evolution. The designer remains in control of the process and whether the outcome of a DTF is acceptable or otherwise.) In this way the PNF enforces rigid DT use but no concurrency is allowed.

The process model corresponding to Case 1 therefore relates to the 'shallow' control issue, by which DT transactions are managed, while the model corresponding to Case 3 relates also to the 'deep' control issue, by which knowledge is introduced in relation to design purpose so that the use of the DTFs is constrained within a specific design session.

The PNF can be changed in mid-process, should it become necessary to adapt the rigidity of the design process. The external Petri-Net description and the dynamic loading makes it easy to change the process being enacted. Note however that at the present time no tools are available for process model design or to check that any new process model is consistent with the current state.

Each node in the Petri-Net corresponds to a a design function and triggers a knowledge predicate which "knows" what should be done at this point in the process, i.e. it handles the internals of the design function. This is where the problematic issue of concurrency is handled. The knowledge base has access to the IBDS' Blackboard (i.e. the design process state) and to the DES (i.e. the state of the problem description). This knowledge base will either be established to react only to the Petri-Net (when in prescriptive mode of operation), or to react to the design process state (when in reactive mode of operation). At the present time these two state are mutually exclusive since they are controlled by preventing the knowledge base from examining the Blackboard's Journal area in the former case. After deciding to carry out the design function, the knowledge base ensures that a) the data required for the task is available, b) starts the appropriate design tool at the appropriate point and then c) hands control to the user. It then monitors what the design tool is doing and finally ensures that the results of the tool are captured and propagated. While the DES is responsible for the handling of the data, the AKH is responsible for driving the process (i.e triggering state changes in the Petri-Net), propagating information to other knowledge bases and keeping track of the design status and history.

The process and design tool knowledge bases are event driven and operate asynchronously. This enables concurrency. Event driven controllers can handle any amount of concurrency, subject only to their ability to "understand" what the other controllers are "saying". In practice, unconstrained concurrency is of little value as it is inherently unstable and unpredictable. The design tool knowledge base is therefore made subordinate to the process knowledge base, which activates/de-activates the former as appropriate. By activating more than one at a time, Petri-Net handling can effectively move from single token passing to coloured token based. Furthermore, design tool knowledge bases can be forced to listen only to the Petri-Net, giving a slavish compliance to the specified process, or encouraged to react to other tools giving a more dynamic, context sensitive system.

IBDS EXTERNAL VIEW

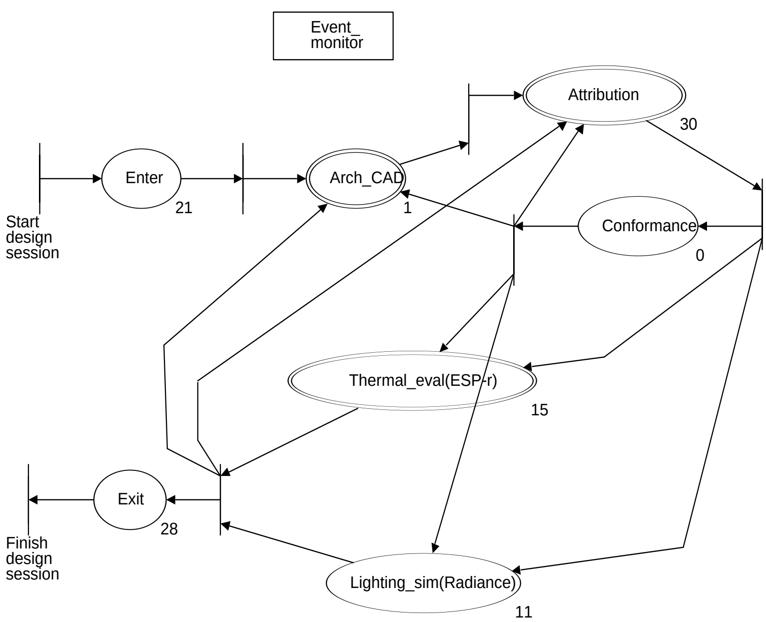


Figure 3: Case 2 process model Petri-Net.

Figure 3 shows the arrangement of the IBDS DTs corresponding to a Case 2 process model (DT invocation sequenced but use unconstrained). On entering the design session the user is required to use AutoCAD to create a new problem geometry (while complying with a set of entity and topological constraints). On exiting AutoCAD, ATTRIBUTE is used to complete the site, composition and operational characteristics of the problem. On completing attribution the user is presented with a choice of compliance checking or thermal/lighting performance appraisal. In the case of compliance checking the conclusions provided will influence a user's choice to modify the problems geometry (via AutoCAD), its composition (via ATTRIBUTE) or invoke either an energy/comfort assessment via ESP-r or a lighting/visual evaluation via RADIANCE. Finally, the user can either revisit AutoCAD or ATTRIBUTE or exit the design session. It is emphasised that although the IBDS supports a cooperative dialogue between the user and the above DTs, this design session, though sequenced, possesses no knowledge about design purpose.

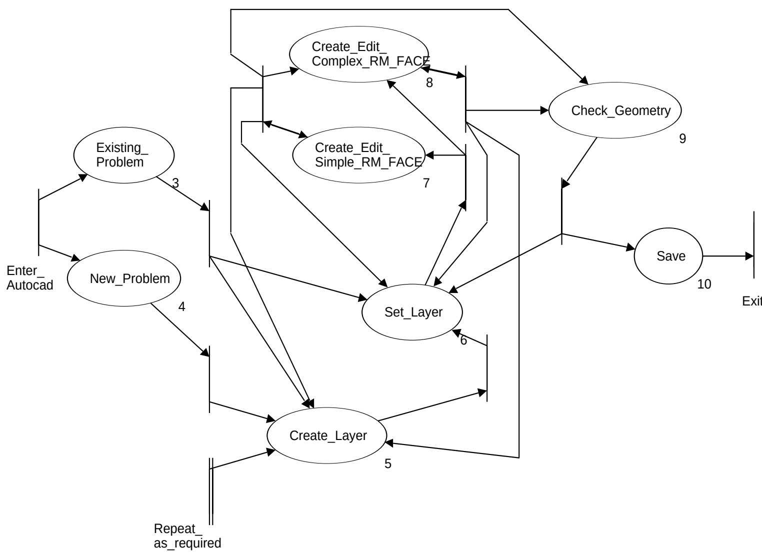


Figure 4: Arch_CAD Petri-Net.

Figure 4 summarises the functionality of AutoCAD as deployed within the IBDS. Because this tool is not "on-line" with the DES, its use must be constrained so that, for example, an isolated line cannot be defined which would have no meaning within ATTRIBUTE.

The result of an initial AutoCAD session might result in a problem representation such as that shown in Figure 5. This contains a simple cubic space bounded on two sides by an 'L' shaped space which includes a window. This geometry is passed to the DES before the ATTRIBUTE DT is invoked.

Figure 6 summarises the attribution phase of an IBDS session. Upon entry to ATTRIBUTE the geometry as described within AutoCAD is recovered from the DES and the following functionality is activated.

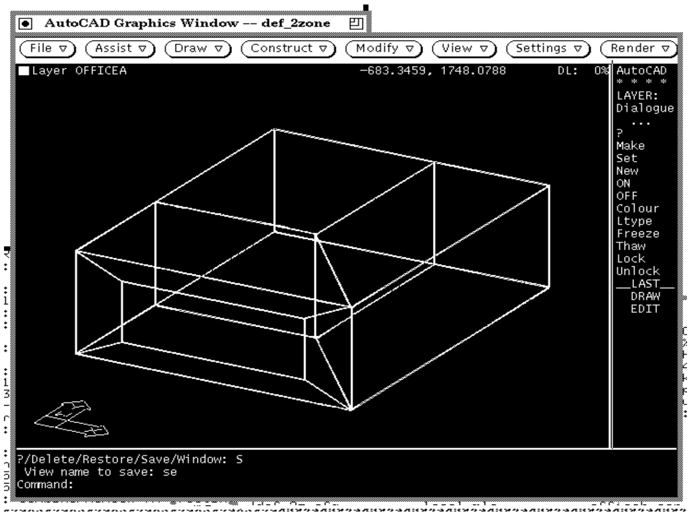


Figure 5: Initial AutoCAD session geometry.

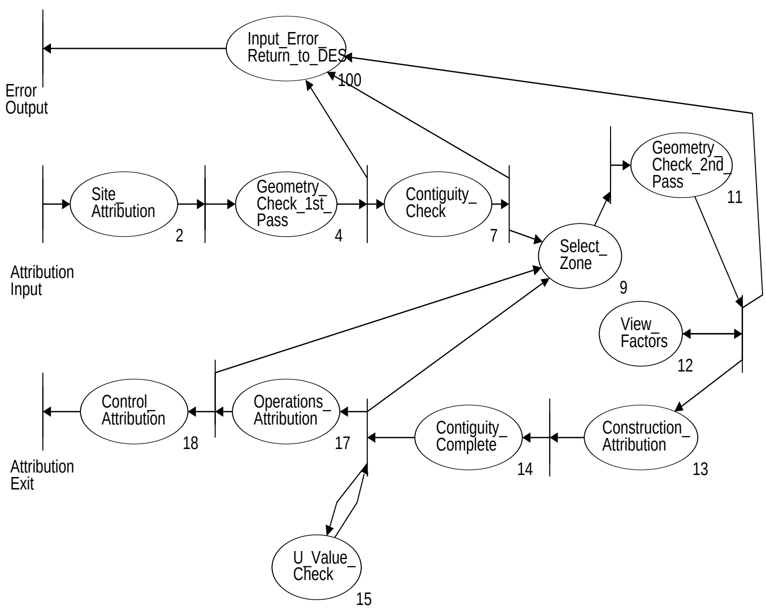


Figure 6: Attribution Petri-Net.

- Description of the site in terms of location and climate.
- Checking of each of the geometric entities for significant errors. If any are found they can be corrected and reported back to the DES.
- Contiguity checking for all zones.
- Constructional attribution of the geometric entities. A typical session is shown in Figure 7.
- Operational attribution by zone.

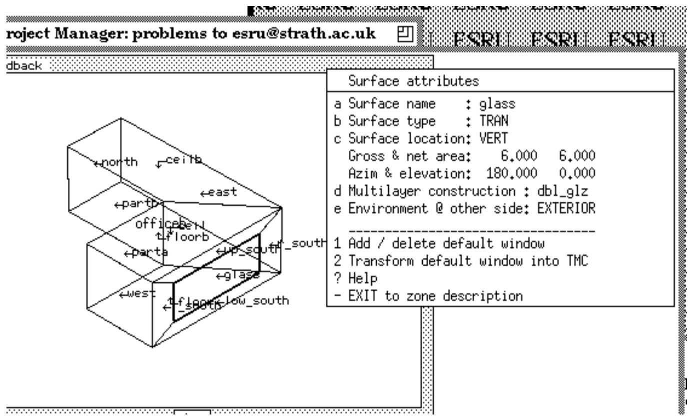


Figure 7: Constructional attribution of a surface.

- Control attribution by zone.
- Zone quantification in terms of areas, volumes, U-values, etc.
- Zone view factor estimation.

After the problem is attributed, the current problem state is returned to the DES. Depending on the complexity of the problem and working preference, the Figure 3 Petri-Net allows the designer to revisit this DT to add further attribution as design information becomes available.

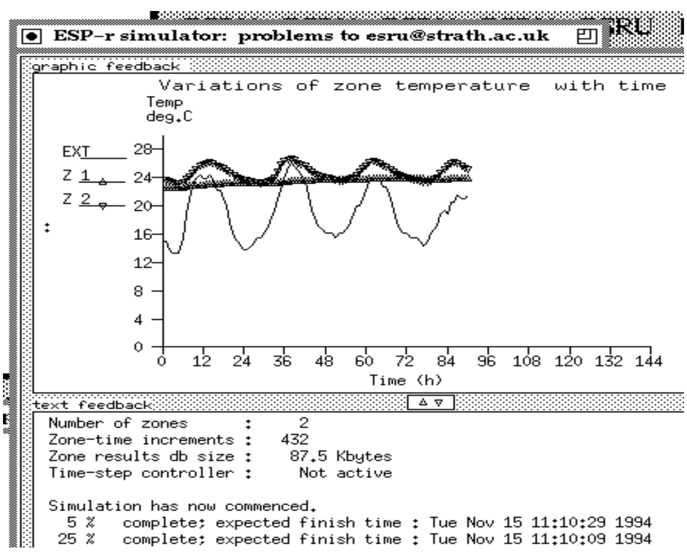


Figure 8: An ESP-r simulation in progress.

Figure 8 was captured during the operation of the thermal evaluation DT and indicates a slight overheating problem.

Finally, Figure 9 shows a typical image as generated by the visualisation DT.

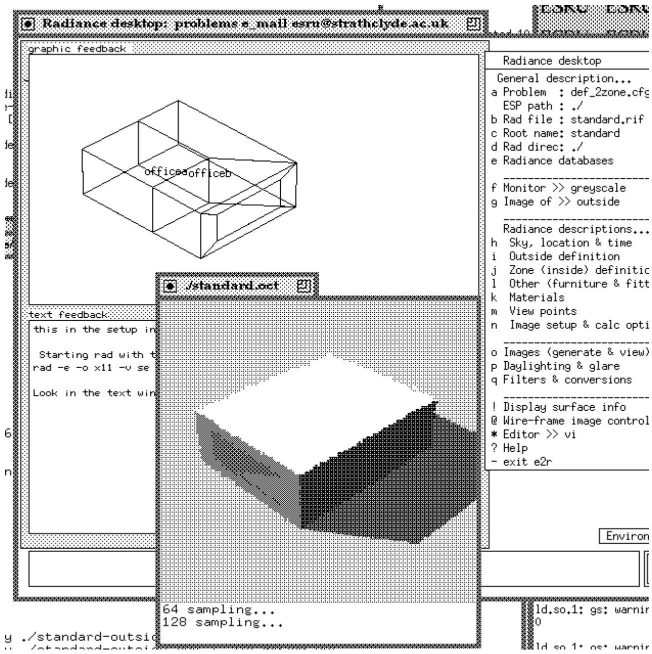


Figure 9: Visual assessment via RADIANCE.

IBDS INTERNAL VIEW

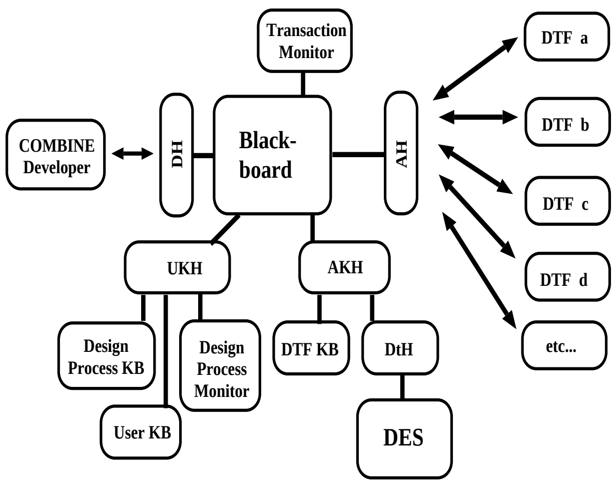


Figure 10: Structure of the IBDS.

Figure 10 shows the internal structure of the IBDS. There are two knowledge handlers, corresponding to user and application control, communicating through the Blackboard areas as indicated. The message passing between the user and application domains has been isolated within a "transaction" area of the Blackboard. The aim of introducing knowledge in relating to design purpose is further supported by the addition of a "journal" area on the Blackboard. This is a repository for the aggregate log of transactions within the system and is used to feed the Prolog predicates of the design session knowledge base. In particular, the nature of the DTs presented to the user, and how they are sequenced and constrained, is supported by the addition of design process knowledge to the AKH. This has been achieved by arranging for the AKH to load the Petri-Net representations as implied by the user's choice of design session.

Between the AKH and the Data Handler/DES resides the Process Monitor which presents the current position of the token in the Petri-Net and the passing of STEP files to and from the DES.

Also shown in Figure 10 is the Transaction Monitor (TM) which observes the transactions between the knowledge handlers, the DTFs and DES. While the TM is an aide to IBDS development it can also be used to observe and analyse an active design session.

In order to explain the working of the IBDS, a series of snap-shots follow which record a user's progress with the active design session corresponding to Case 2 as outlined previously. In the snap-shots the arrows show the potential flows of information: a single arrow indicates a notification while double arrows indicate sending and listening. The "user_dialog" area of the Blackboard is reserved for user interaction transactions, while the "application_dialog" area is reserved for transactions related to the DTFs. The "journal" area receives messages from the various knowledge handlers and organises these for subsequent analysis and process control.

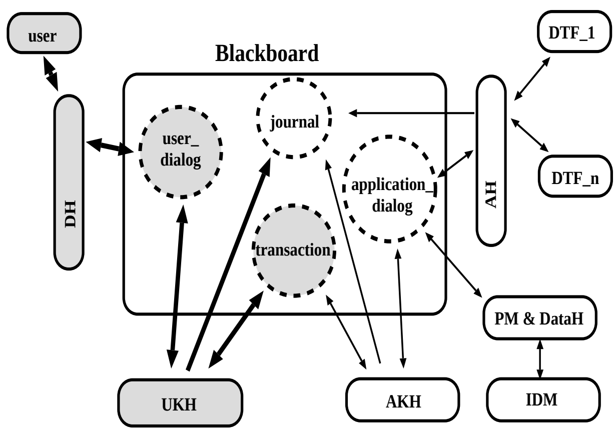


Figure 11: State of the IBDS after user action.

Figure 11 defines the state of the system after the user has selected a DTF and the actions triggered:

1) a message passes to the Dialogue Handler (DH) indicating the requested interaction;

2) the DH passes the message to the "user_dialog" area;

3) and the User Knowledge Handler (UKH) tells the Blackboard to "start DT".

Figure 12 is the state of the Blackboard after the UKH has issued a message "application_dialog start DT" to the transaction area. The Application Knowledge Handler (AKH) finds the actual application and posts the message "new_application DTx" to the "application_dialog" area. The Data Handler

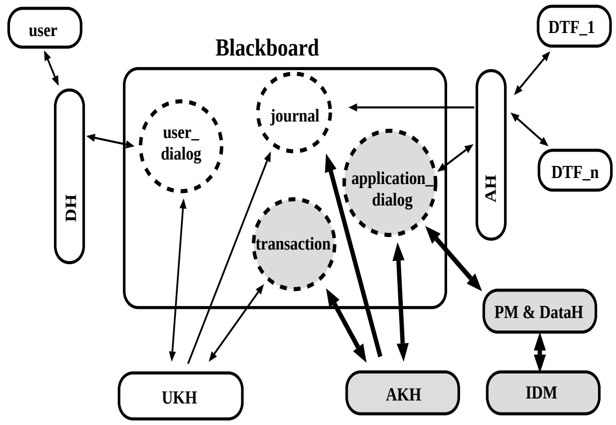


Figure 12: User knowledge handler requesting a DTF.

(DtH) then queries the DES - "get_data_for DTx" - and the DES returns the appropriate data for DTx as "data_for DTx file".

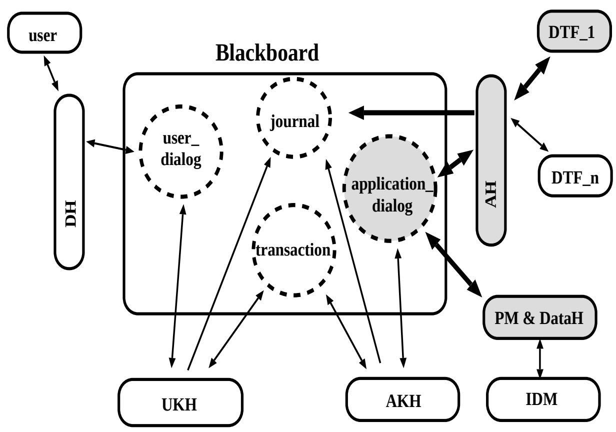


Figure 13: DES returns STEP file and DTF starts.

In Figure 13, after the DES issues "data_for DTx file":

1) The DtH posts "new_application application parameters" to the application dialog area.

2) The Application Handler (AH) starts the application and establishes a pipe to receive the performance return(s).

3) When the application is complete the AH records this and sends "closed DTx revised_data_file" to the application_dialog area. The AKH posts "closed DTx" to the transaction area, which is receives by the UKH for transmission (not shown) back to the user interface.

INTRODUCING DESIGN PURPOSE

Consider now a session which incorporates knowledge in relation to design purpose (Case 3). With reference to the Petri-Net in Figure 14, the initial portion of the design session might proceed as in the previous unconstrained session in terms of geometric specifications and attribution. For those users who enter the summer overheating design session with an existing problem there is a direct path to the overheating assessment.

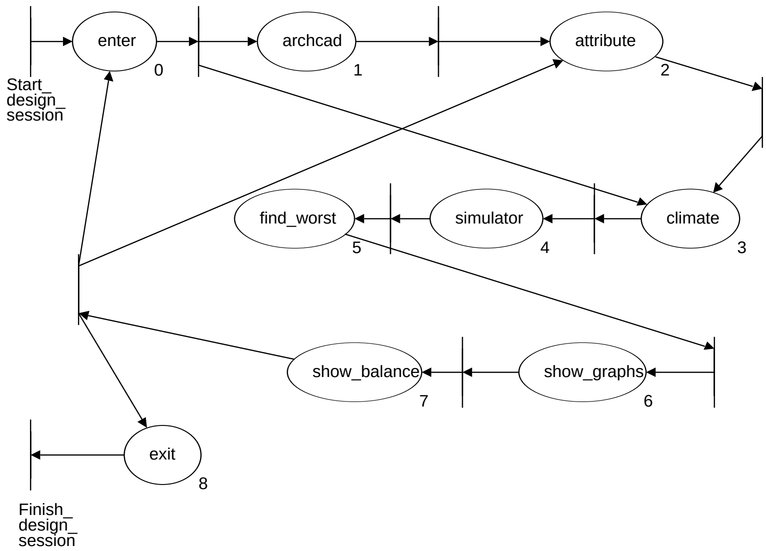


Figure 14: Summer overheating Petri-Net.

Simulation environments, such as ESP-r, have traditionally provided facilities to enable direct access to their internal DTFs. In the current example some of these ESP-r DTFs are accessed, with the user involved as the arbitrator of acceptability or otherwise of the performance returns. Here, the process model involves the determination of the climate patterns which would constitute an acceptable test of summer overheating risk. While such a decision is implicit in most simulation based studies, here it has been made explicit. Next, the focus is shifted to a simulation of the current problem and then determination of what constitutes the worst zone in the problem. The search rules operate on the basis of the highest resultant temperature in an occupied space as determined from a series of inquiries of the database of simulation results. It is quite possible to have alternative rules governing this DTF.

Assuming that overheating has been detected, two presentations are made to the user. Firstly, a frequency binning of temperatures and a graph of temperatures in the worst zone. This sets the context which violated the 'rules' of the assessment. The process model then calls for the presentation of information on the likely causal factors. For example, if high internal gains were the cause of the overheating then only this information would be provided. The user can then either exit the design session or return to the architectural CAD or to the ATTRIBUTE DTF. A typical design session is shown in Figure 15.

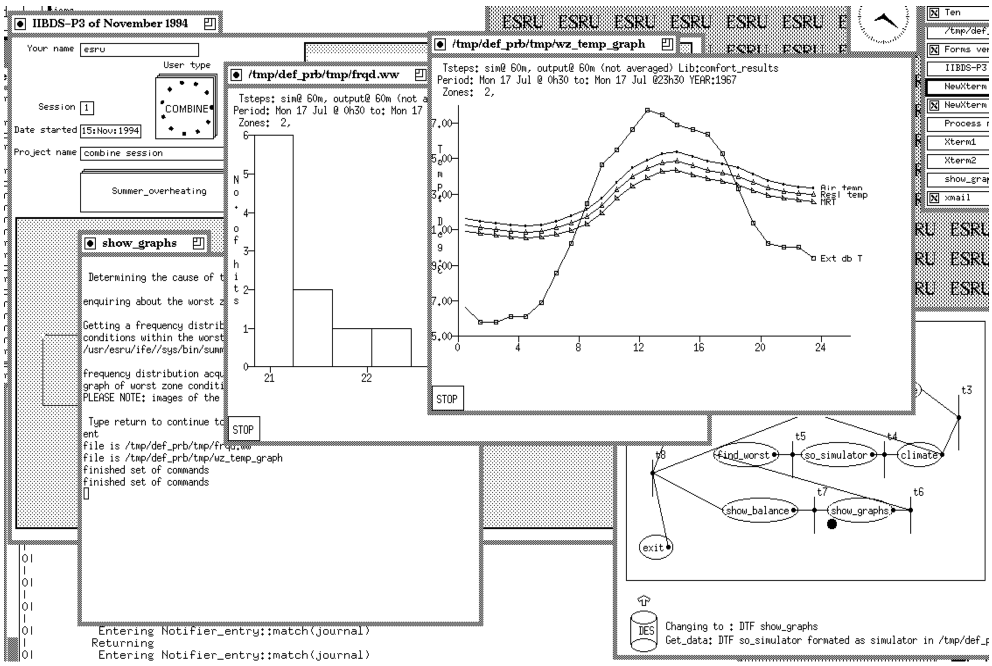


Figure 15: Summer overheating design session.

CONCLUSIONS AND FUTURE WORK

Technology has now reached a stage where it is possible to bring together product modelling and intelligent interfaces. The COMBINE project has undertaken developments in these areas, resulting in an Integrated Data Model (IDM) and Intelligent Integrated Building Design System (IBDS). The former is able to service the building description requirements of several disparate design tools; the latter is able to control the operation of these same tools against rules which define the purpose of their use. Taken together, these developments are helping to evolve the prospects for a Computer-Supported Design Environment by which the analytical power of the computer can better complement the creative power of the designer.

REFERENCES

Augenbroe G., ‘Integrated Building Performance Evaluation in the Early Design Stages’ *Building and Environment* **V27 N2** pp149-61, 1992.

Clarke J., *Energy Simulation in Building Design*, Adam Hilger, Bristol and Boston, 1985.

Spiby P. (Ed), ‘EXPRESS Language Reference Manual’ *ISO TC184/SC4/WG5 Document N14*, 1991.

Ward G., ‘The RADIANCE 2.3 Synthetic Imaging System’ *University of California* Berkeley, 1993.

Autodesk Ltd, ‘AutoCAD Release 10 Reference Manual’ *Autodesk Ltd.* Exeter, 1989.

Hand J., ‘Enabling Project Management Within Simulation Programmes’ *ESRU Pub T94/14* 1994.

Rode C.,‘Specification of the Generic Tool: BRC, the Building Regulations Compliance Checker’ *Danish Building Research Institute* Horlsholm Denmark, 1993.

Javor A ‘Petri Nets in Simulation’ *EUROSIM - Simulation News Europe* November pp6-7 1993.