THE ESP-r COOKBOOK

Strategies for Deploying Virtual Representations of the Build Environment

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

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

4 September, 2008

This document is designed to be printed in one of the following formats:

Two pages per sheet of A4 or B4 paper

At 85% scale on A4 paper (wide margins for notes)

At 100% scale on A5 or B5 paper

It is designed to take less space on the screen than the previous edition.

Table of Contents

ABSTRACT

This *Cookbook* uses the general purpose simulation suite ESP-r as a platform to explore *strategies* for deploying virtual representations of the built environment to answer questions posed in the real world of design and research groups.

The *Cookbook* talks about translating client questions into virtual representations that are no more and no less complex than is required for the task. It talks about re-discovering the power of pencils and paper and it dares to mention the word methodology. And discovering valuable patterns in the clutter and then learning the art of responding to what if questions. And since the author is professionally paranoid you might pick up some new definitions of the word QA.

Almost all of the *strategies* presented can be applied to the task of creating elegant virtual representations in other simulation suites. Readers might allert their colleagues to take a peak.

ACKNOWLEDGMENTS

This book could have been completed only within the exceptional group environment of the Energy Systems Research Unit of the University of Strathclyde in Glasgow Scotland. Where else could an architect compose tens of thousands of lines of source code and then use the resulting virtual edifice to explore and support the design process and then turn the process on its head to return to the written page to explore strategies of its use.
The author would also like to acknowledge the time which Samsung Construction allowed the author during a period of secondment in Seoul.

Chapter 1

INTRODUCTION

1 Introduction

Learning how to use simulation tools for design decision support and for research has tended to follow three paths - the mentor path, the workshop path and the *there-be-dragons* path. The mentor path works exceedingly well and is an efficient, if not particularly inexpensive way of gaining the skills and tactics needed to apply simulation to real-time projects.

Workshops are another successful approach to simulation training. Two or three days of initial sessions, supplemented by advanced topic workshops and the occasional email allows many practitioners to productively use simulation. Both of these approaches rely on personal contact with an expert and iterations of demonstration, followed hands-on experience and dialog for skills acquisition.

Many practitioners rely on mentoring and workshops to keep them up to date as tools evolve and for exploring new facilities. Documentation tends to lag the evolution of simulation tools and many lesser-used tasks may not be well documented or documented in ways accessible only to geeks.

What you are reading now (and the companion *Cookbook Exercises* is addressed primarily at those who are taking the path of confronting the dragons. It has also been used to support workshops in conjunction with the exercise volume.

The *Cookbook* strives to be generic in its discussion. As the title suggests, where specific examples are needed they are based on the ESP-r suite. Some blocks of text apply only to ESP-r and occasionally you may notice the following icon...

If you are reading this from the point of view of another application skip down a few paragraphs.

The *Cookbook* also includes sections of interest to technical support staff and uber-geeks. These are marked with the following icon...

If you have downloaded one of the pre-compiled ESP-r distributions for Linux (most distributions), Mac OSX, Windows (native GUI), Cygwin (emulation under Windows) from the ESRU web site <http://www.esru.strath.ac.uk> or acquired the source from the source code control repository via the uber-geek command (on one line):

```
svn checkout https://esp-r.net/espr/esp-r/branches/development_branch
```

and compiled your own version. Most of the instructions needed to get a working distribution can be found on

the download page. Additional instructions are included with the source and there are discussion lists that might provide additional clues.

And the ESP-r download pages do not really tell you much about what to do once you have ESP-r on your computer. This statement is probably applicable to most other vendors. Of course there are web based tutorials and exercises as well as manuals that approach telephone directory proportions.

Most vendors went through a phase where they *believed* that web based tutorials would supplant mentoring and workshops. From the author's perspective, web pages work less well than the mentor/workshop paths. The *Cookbook* is an attempt to bridge this gap. It evolves, as does ESP-r itself, from observations of practitioners who are attempting to support real-time design assessments of real-world issues.

Simulation tools almost always arrive with a range of example models of two broad types - abstract models which are composed so as to illustrate semantics and syntax and those models derived from consulting projects which focus on specific building performance design issues. The first type is often used by novices to get used to the simulation tool, the second type for those who are looking for examples of best practice models. Vendors do not always make clear which is which.

Example models contain a wealth of information for those who *know what they are looking for*, for those who are *persistent* or for those who are using them as *reference materials* within the context of a workshop or in mentor based training. For these users example models can act as:

- a mechanism for exploring the tool (e.g. where do I find out information about environmental controls, the composition of walls)
- to explore the sequence of tasks required to run an assessment and recover specific performance metrics
- to explore incremental changes in the description of the model and the performance implications of such changes

Creating a model from scratch under close supervision and with commentary on the approach taken does reduce the frequency of *encounters-with-dragons*. In ESRU workshops almost all participants first model works correctly the first time it is simulated.

What you are currently reading expands on existing workshop materials and years of mentoring, recast for the resolution of the printed page.

The goal is not simply to act as a dictionary or reference but as a guide to how to approach realistic design decision support in real time, delivering real information and still have time for a cup of coffee at the end of the day.

This document is based on the premiss that readers will already have an intuition about the physics of buildings and environmental systems. A future revision is planned for those who are less opinionated. And for readers who are users of other tools there will be much of value even if the details of implementing the methods differs. Who knows, someday there may be an

EnergyPlus Cookbook and an *EE4 Cookbook*.

A word about ESP-r versions

ESP-r is under active development. On any given day there may be a half dozen commits of code or documentation or updates to exemplar models to the repository. This *Cookbook* evolves at a slower pace. This 2008 version has been revised to match the evolved interface of ESP-r but that match is likely to be imperfect. Interface entities and paths such as `Model Management -> browse/edit/simulate -> composition -> geometry & attribution` may have a different syntax. If you don't find a match, please look around for something similar.

You may also notice is that some interface related figures in the *Cookbook* look different from what you see on the monitor. There are currently three different interfaces for ESP-r. There is the traditional X11 interface which has its roots in the world of UNIX and Linux. There is an almost complete port of ESP-r to a graphic library called GTK. GTK is implemented on a dozen operating systems and this allows ESP-r to be run as a native Windows executable. It also has a more familiar look and feel and once the port is complete it will be the primary interfaces to ESP-r. The third interface is a pure-text interface which tends to be used for scripted production work or to enable ESP-r to act as a background engine for other software. Look in the *Version Appendix* to see typical dialogues from the different interfaces.

With the exception of file browsing facilites, the command sequence needed to undertake most tasks is almost identical across each of the interfaces. Where facilities differ you may see one of the following icons followed by specific instructions...

1.1 Tactical approaches

Depending on your personal preferences, getting acquainted with a simulation tool either begins with exploring existing models (in ESP-r these are called exemplars) or in the context of creating a model from scratch.

If you are taking the *from scratch route* grab a note-pad and some sketch paper. The following sections explore how you can use ESP-r to arrive at a working simulation model and a growing set of simulation skills. If you are using a different simulation suite keep reading. *Tactics can almost always be applied universally*.

Lets begin by deciding what kind of model we are going to make and then *plan the work* so that it fits our resources. This is a *tactical approach* to simulation which concentrates on the art of making concise models to answer our clients questions without delaying the design process.

The first table is a powerful dragon slayer. Clients ask us questions - but what are the *questions we ask ourselves* as we plan and then compose our virtual worlds?

Table 1.1 Initial tactics:

Design Question	Simulation questions
What do we want to know about the design?	What thermophysical issues should be addressed by the model?
How do I know if the design works?	What performance can I measure to inform my judgments? What level of model detail is required for this?
How might the design fail?	What boundary conditions and operating regimes would be a reasonable test?
How do I match the information I have with the requirements of the tool?	What is the essence of the design in terms of form, composition, operation and control? What essential interactions need to be represented? What facilities can be employed and what skills are needed to use them?
Is our approach ok?	Can I sketch out my model and explain it to others?
Are the performance predictions credible?	What assessments need to be undertaken to gain confidence in the model? What is expected of a best practice design?
How can I deliver the most value for my client?	What else would clarify how the design works? How might the design and the model evolve during the design process? What would I do now to make it easier to work with this model again after a four month delay?

Without tactics we will miss out on the value-added aspects of simulation which cost us little to implement but deliver substantial benefits. A tactical approach keeps you *in charge* of the simulation tool.

Simulation models have a context within which they are created and evolve. In the next section the clients specification and design questions form the context. From this we decide what type of model(s) the specification implies as well as the assessment(s) that need to be done to answer the clients questions or further our research goals. The plural is intentional - real projects are iterative and models either evolve or spawn the next generations of model.

1.2 The client specification

The following section provides the specification of our first project. It is designed to allow an exploration of the best-practice choices made while planning simulation projects.

No matter what simulation tool you are using, there is always more than one approach to a given task. Workshops typically use sequences that are known to work. Mentors will encourage you to explore alternative approaches. Enlightened managers allocate time for such explorations.

This initial simulation project is part of a general practitioner's office. The client specification is intentionally terse so as to demonstrate typical decisions made by simulation teams in practice. Clients have beliefs about how buildings work and simulation is one approach which can be used to confirm or refute such beliefs.

Figure 1.1 shows a plan and section (looking from the east) of the general practitioers office. The reception has a flat roof and the examination room has a sloped roof with a skylight to the north.

Figure 1.2 is a wireframe perspective view (looking from the south-west). Note the strip windows on the north of the reception and the two strip windows on the south facade.

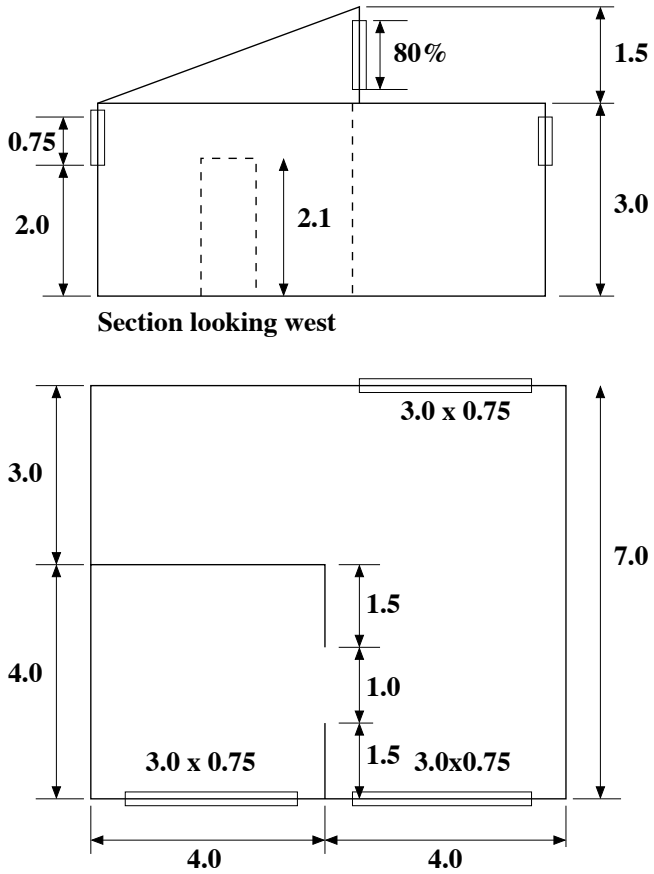


Figure 1.1 Plan and section of general practitioner office.

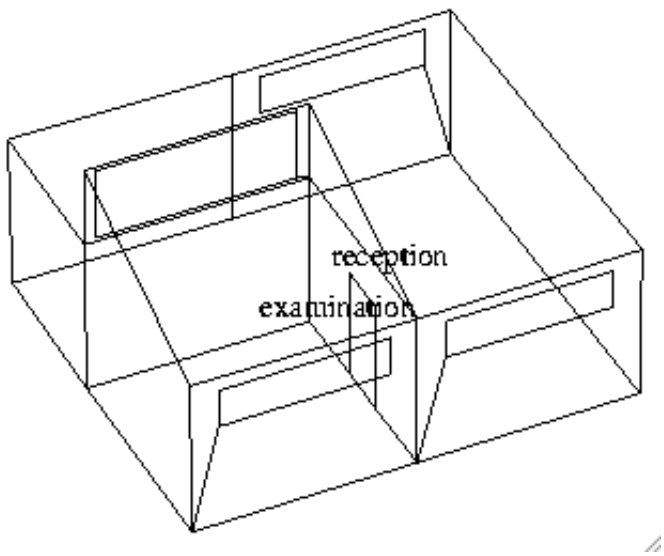


Figure 1.2 Wire-frame view of general practitioner office.

Figure 1.3 is a colour rendering (looking from the south-west) which was created by exporting the ESP-r model

to Radiance.

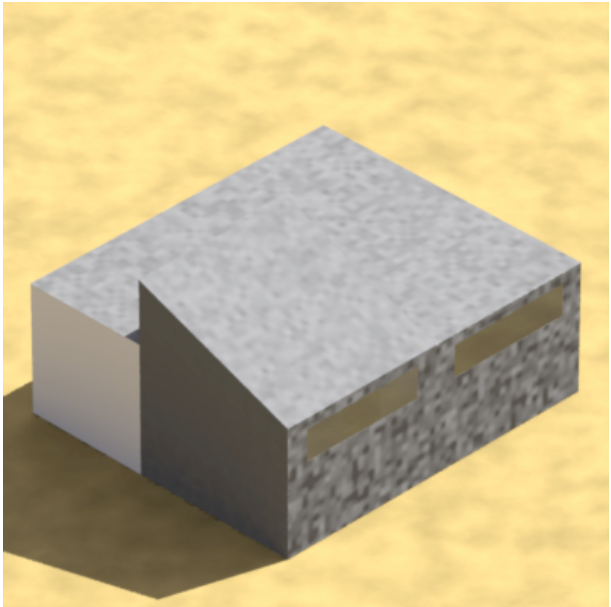


Figure 1.3 South-west view of general practitioner office.

Figure 1.4 is a view from the north-west through the examination room. The white surfaces represent walls which are partitions to a portion of the building which has not been included within the model.

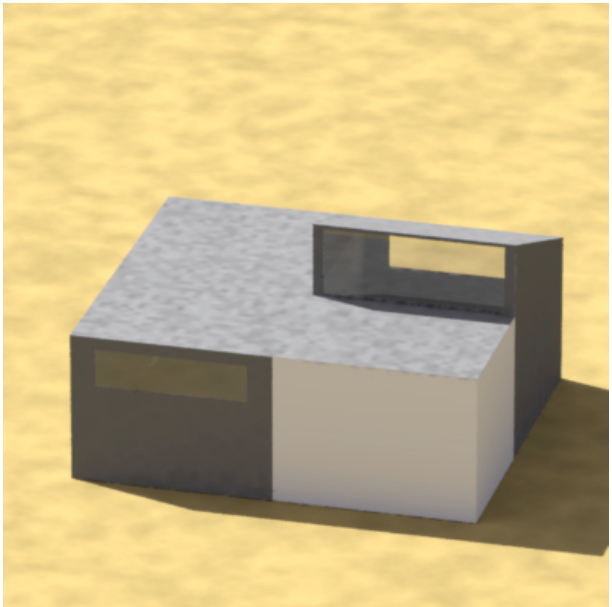


Figure 1.4 North-west view of general practitioner office.

This project represents a **portion** of a general practitioner's office. Focusing initially on a portion of a building is a powerful strategy and one which is applicable to almost all simulation tools.

The client indicates that this medical practice has a brisk turn-over of clients and that, on average, there are two people in the examination room during the hours of 9h00 to 16h00 on weekdays (200W sensible, 100W latent). The reception area serves other portions of the building which are not included in this model and there might be up to five people. Lighting in the reception is 150W during the hours of 8h00 to 19h00 and there are no small power loads in either room for purposes of this model.

The heating set point is 20°C and the cooling set point is 23°C between 9h00 and 17h00 on weekdays with frost protection (15°C) on weekends. The client has no specific opinion as to how this is to be achieved.

ESP-r, unlike some simulation suites, includes both 'ideal zone controls' and component based descriptions of environmental systems. In this exercise we will start with a minimalist 'ideal' description and assume that both heating and cooling are assumed to be delivered convectively. ESP-r demands an initial guess at the heating and cooling capacity, but otherwise we will maintain our focus on demand side issues.

Questions to ask about *auto-size* functions:
a) what boundary condition(s) and operational regime(s) are associated with peak conditions?
b) what method(s) are used to assess intra-component dependencies as components are sized?
c) what criteria are used to determine which sub-optimal set of component sizes works best?
d) what criteria might you use to confirm the suggested sizes?

Components might *seem* unambiguous. Be sceptical until you can confirm that they match your expectations.

Back to our initial model. Even the best of buildings have infiltration. There is a discussion about air flows in a later section. For now lets use an initial engineering assumption that there will be 0.5 ac/hr infiltration at all hours.

1.3 Design questions

The client wants to know what the typical demands for heating, heating capacity, thermal comfort in the winter and summer, whether it is likely to overheat and if the daylight distribution is ok.

To answer these questions we require a model which represents the general form, composition and use as described in the client specification. The model need not be particularly detailed and our goal is to maintain the volume of the spaces as well as the orientation, area, distribution of mass and general shape of the room surfaces.

Review Table 1.1. If the client asked different questions the nature of the assessments might well be different.

So, what sort of assessments will address the question of typical heating and cooling demands, capacity and comfort? If we weren't thinking tactically we might run an annual simulation and then get bogged down in scanning the predictions for useful information.

A *tactical approach* limits the quantity of information we have to deal so both the model and its performance is easier to *understand* and the *QA burden* is reduced. Lets look first on seasonal patterns to highlight performance issues. Computers may process a year in seconds but *QA staff costs* are greater.

The key initial objective is to support our own *understanding of performance* by looking at patterns in a limited set of data and so be able to spot glitches in our model as well as opportunities for improvements to the design (or the clients specification) as soon as possible.

Value added: The client did not ask for it, but it takes little extra effort to check

for typical spring and autumn performance might provide useful feedback to the design team.

Another tactic is to define *performance metrics* (e.g. what can we measure in our virtual world) early in the process. Some metrics e.g an energy balance within a zone, might contribute to our own understanding of the design and other metrics e.g. thermal comfort might be useful to report to others in the design team.

ESP-r workshops typically devote as much time to exploring building performance issues as is spent on model creation. Simulation suites which do not include an interactive exploration facility will include a descriptive language to specify what performance metrics are to be captured during each assessment - so learn that language!

The metric for heating and cooling demands is kWhr (integrated) over the week and for capacity the metric is diversified kW (the peak capacity required for this portion of the building). Just to be sure that the pattern of demand is reasonable we will want to graph this. In addition to a table of demands and capacity we might include the graph in our report if it proves of interest.

Resultant temperature is a common comfort metric. A frequency bin of resultant temperatures during the occupied periods would inform the client about the distribution of comfort. For our own use, we also want to check the number of hours over 24°C and graph the temperatures, we might include these in our report if they prove interesting.

To answer the question about daylighting we can look at daylight factors across a grid in each of the rooms. To build a model that will answer questions of thermal and lighting performance we need to decide how much geometric resolution if required. In the case of daylight factors the level of detail needed for the thermal assessment should suffice. If glare was to be assessed the model would need to include additional visual geometric details. Later on we will consider tactics that anticipate probable future design performance questions.

> QA tip: Write down these decisions, we will want to review them as the project progresses to make sure we are working to-the-plan.

1.4 Model planning

Get out your grid paper and note pads and keep the laptop lid closed for now. Pre-processing information and sketching the composition of our model will limit errors and make it easier for others to understand what we intend to create, and, after we have made it, to help check that it is correct. This rule applies whether we are going to import CAD data or use the in-built CAD functions of our simulation tool.

It also saves time and removes another source of error if we convert the important horizontal and vertical dimensions (such as those shown in Figure 1.1) to model coordinates and include them in our planning sketches. This avoids jumping between a keyboard and a calculator during model definition as well as helping in QA tasks

User friendly software **does not** reduce the need for planning or robust QA. If anything, it is even more important to guard against needless complexity. Lets call this *avoiding the dark side*.

An exercise related to model plannning is included in the *Cookbook Exercises* within Exercise 1.

1.5 Model Coordinates

Novice practitioners often proceed under the assumption that geometric input consumes the bulk of their project time. Seasoned simulationists know that geometry takes about a third of their project time and they evolve strategies to help them limit the time spent on creating and checking geometric entities (so they will have the time and attention to leverage value-added opportunities that might arise).

An experienced user can generate models with scores of thermal zones that fit together correctly the first time. Such skills can be acquired over time. We are going to walk before we run, and our initial goal is to create a correct three zone model for the doctors office. In workshops nine out of ten participants create models which simulate correctly the first time. If asked, most are able to re-create this model with minimal support and in 25-35% less time. So even though an experienced user will outpace a novice, good working practices ensure that even novices can produce useful models.

The approach we take to create the form of the model is as dependant on the *questions* we wish to address with the model as it is on the specifics of the *building blocks* and input facilities that are offered by the simulation suite.

- questions about general comfort and energy demands at peak and moderate climate conditions require only a moderate geometric resolution e.g. correct volume of the space, approximate location of doors and windows
- questions related to comfort at a specific location require higher geometric resolution, especially if surface temperatures are likely to be vary across a surface
- question related to visual comfort will require higher geometric resolution for facades and may require that furniture within rooms and outside obstructions be accounted for
- questions related to the distribution of air temperature within a physical space may require that it be represented by more than one thermal zone or that it include a CFD domain
- questions related to passive solar performance may require a higher level of geometric and construction detail to assess the impact of mass and the distribution of solar radiation

Each of these issues require that we first consider the physics underpinning the assessment. Second we must search the available model building blocks for relevant entities. Lastly we must consider what resolution to apply those entities within our model.

For example, a passive solar design will be sensitive to heat stored in the fabric of the room as well as details of glazing in the facade and in partitions to adjacent rooms. The surface temperatures in a sun patch might be substantially elevated. To find out where the sun falls in the room at different times of the year we might create a rough model and then check what we can see in a wire-frame view at different times of the year. Our goal would be to find out if we need to subdivide surfaces to better reflect the temperature differences in insolated and uninsolated portions. We can then make a variant of the zone with higher geometric resolution and compare the predicted surface temperatures.

As a general rule the design of models should ensure that the volume of the air is close to the correct value just as we want to ensure that the surface area is correct and that mass within the rooms is appropriately distributed.

In the doctors office the windows are not large and the questions are general and so the exact location of the windows is not critical (but it costs us nothing to place them accurately).

The dimensions shown in Figure 1.1 should be straightforward to represent. Looking closer, there is no thickness indicated so the criteria used to arrive at the dimensions is unclear. If you were tasked with determining dimensions from information supplied by a client a set of rules would be useful.

- Where the volume of the space is large with respect to the thickness of the facade and where the complexity of the facade is low it is common to measure from the inside face of exterior walls.
- Measuring partitions at each face is common where the coordinates are taken from CAD drawings and at the centre line during the sketch stage.
- Ceiling voids or raised structural floors with little or no air movement are often represented as layers of air in constructions. Where air movement is likely or there are significant heat gains within the void they may be better represented as a separate thermal zone.
- As ceiling (below) to floor (above) distances increase it makes sense to take the height co-ordinates literally and geometrically separate levels within buildings.

There are ESP-r exemplar models which have the zones on each side of a partition in the same plane and other exemplars have zones separated in space. In most simulation tools it is a matter of personal preference because the heat flow between zones is established by specific directives *partn_cor in office* is connected with *partn_off in corridor* which are separate from the geometric definition.

Most simulation tools represent geometry as polygons and separately represent their composition. Wire-frame displays often present models as having walls of little or no thickness. For modern (thin) construction the wire-frame display may provide an image that allows us to forget that real walls have thickness.

Consider the modern office construction in Figure 1.5 there will be little or no change in predictions whether the centre line or the actual location in space is used. Given that there are doors in the partition there is a strong case for adopting the centre line to avoid visual confusion.

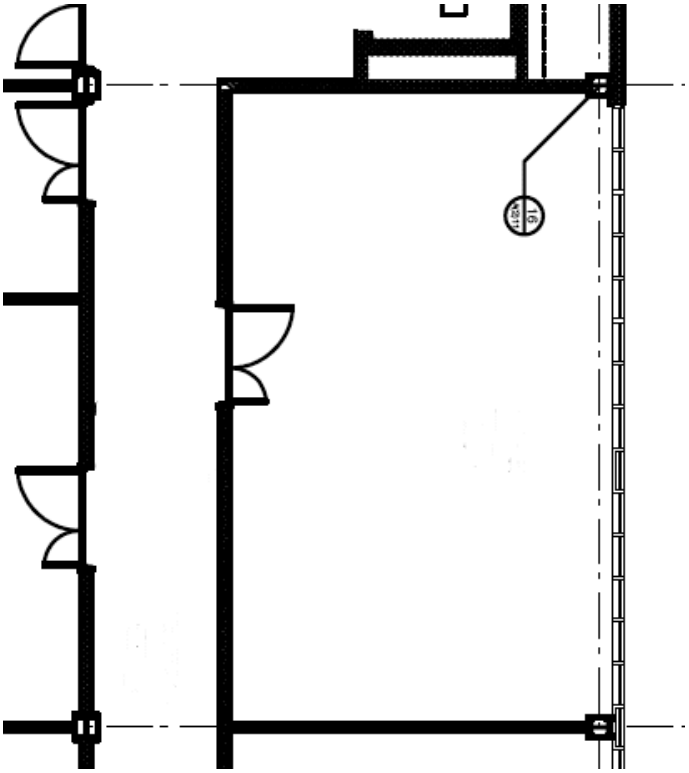


Figure 1.5 Plan of a modern building.

At the other extreme, historical buildings can have exterior walls and partitions which vary in thickness and are substantially different from the thickness of doors. In Figure 1.6 the inside and outside faces are multi-faceted the shape of the window surround influences the distribution of light within the room. Some partitions are thin enough to be treated as centre lines and others suggest a separation of the thermal zones.

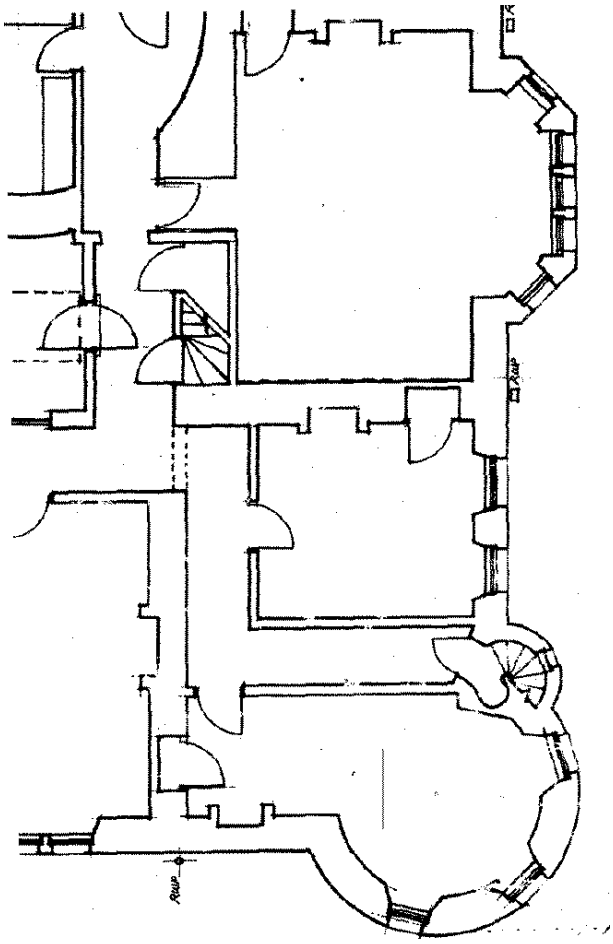


Figure 1.6 Plan of a historic building.

Translating the historical plan into a model required a number of decisions to represent in one dimensional heat flow paths a building which is a substantially three dimensional heat flow problem. The result, shown in Figure 1.7, substantially retains the volume and positions of the spaces rather than the exterior form of the building. Thin partitions are taken to the centre line. Some plan detail has been omitted and minor spaces amalgamated into adjacent rooms.

Having sketched on an overlay what we wanted to transcribe, the actual creation of the initial extruded form of the rooms was accomplished in a matter of minutes via a click-on-bitmap facility.

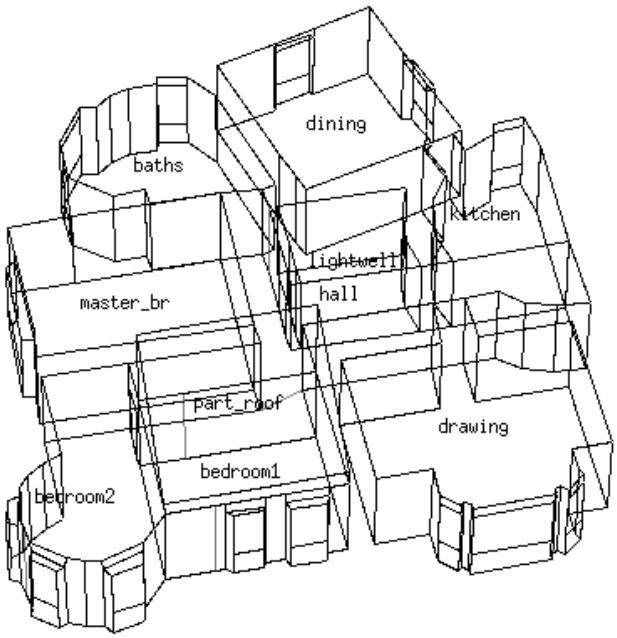


Figure 1.7 Model of a historic building.

The geometry at the window heads and sills was then adapted and the doors inserted. When attributing the surfaces, the associated construction was selected to account for the local cross section.

A further discussion about options for interpreting complex three dimensional designs into appropriate models can be found in Chapter 4.

The *Cookbook* is concerned with the art of composing models which are specifically adapted to the needs of the design process. Not all projects are as demanding as the historic building. There is also an art to creating models which are fit for the sort of general questions posed in the doctors office. While planning a model we might ask ourselves:

- would patterns of temperature and heating change if the volume of the space was off by the width or a wall?
- would more sunlight enter the room if a window was lowered by 5cm?
- is it necessary to include the frame of the window?
- is it necessary to include the furniture within the rooms?

Essentially our concern is to ensure that the uncertainty in the model is constrained to the point where it would be unlikely to change a design decision. Each of the above bullet points could, in fact, be tested by creating model variants and then looking at the performance differences. There are many simulation groups who have undertaken such parametric studies to arrive at their *rule set*. For this initial exercise the rule is *keep it simple*.

The X axis in ESP-r is towards the East and the Y axis is towards the North. Most users find it convenient to keep their model in positive coordinates and to define their model using cardinal orientations and later rotate and transform the model to reflect conditions at the site.

Figure 1.8 shows critical coordinates (X,Y) derived from Figure 1.1. To simplify our task let us assume that the origin of the model is at the lower left corner of the examination room. The critical vertical points to record on your notepad are 0.0 (ground), 2.0 (window sill), 3.0 (ceiling), 4.5 (top of sloped roof).

Taking the time to gather and confirm critical co-ordinates in the plan **and sections** before going to the keyboard is a key technique in getting-it-right-the-first-time.

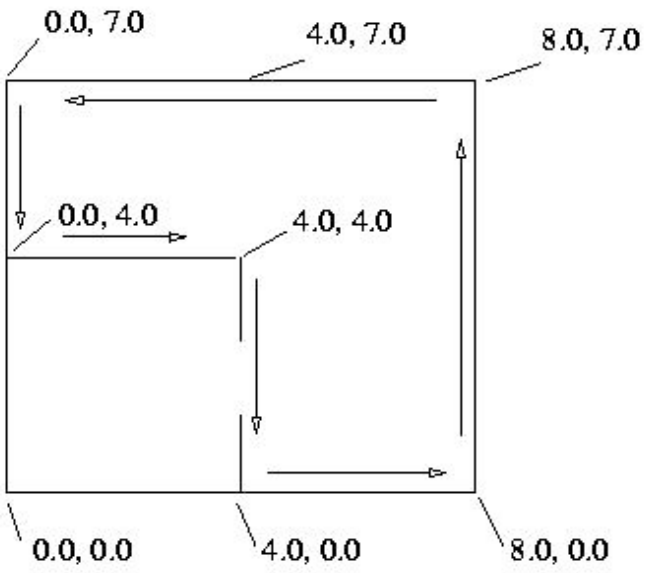


Figure 1.8 Critical dimensions taken from the plan.

1.6 How the building is used

Our next stage in the planning process is to deal with how the building is used (schedules of occupants, lighting, small power). The client specification must be transformed into schedules. Experienced user will either sketch the day schedules or record the time and values for each casual gain data for each day type just like they did for the co-ordinates.

Just as defining an appropriate level of geometric detail is important, schedules can be crafted to test a number of performance characteristics within a single assessment. Why bother? Because a few minutes effort can give early clues of how buildings may fail and how the building fabric and its systems respond.

Another reason to bother is that "all staff are here all the time and copy machines have a constant queue of telephone directory length reports" is not usually how buildings are used. It might be a secondary question to ask when testing risk to have such an extreme as an alternative, but not as the primary operational regime.

The examination and reception spaces have a simple schedule of occupancy which includes some diversity. For example, there is a lunch hour and there is a ramp-up and ramp-down of gains at the start and end of the day to represent cleaning staff in the morning and stragglers at the close of work. In both cases there are periods during the morning and afternoon with full loads so that capacity issues and the potential for overheating are addressed.

ESP-r represents internal (casual) gains as a schedule which applies to each defined day type. By default there are weekdays, Saturdays and Sundays. For this exercise lets stick with a default set of day types.

You can lump all casual gains together for definition and reporting purposes or use up to three separate types of casual gains. Typically the first type is for occupants, the second is for lights and the third is for small power (equipment).

Each of the days has one or more periods associated with each type of casual gains. Periods must not overlap and

should cover the entire day (0h00 to 24h00. Each period has a sensible load (W), a latent load (W) as well as the fraction of the sensible load which is radiant and convective.

In the reception occupant sensible gains are 80W from 7h-8h, 240W from 8h-9h and 12h-14h and 400W from 9h-12h and 14h-17h. For purposes of this exercise occupant latent loads in the reception are assumed to be half the sensible loads. Consider though what might be happening in such spaces. What is the latent gain from several cups of tea or a boiling kettle?

For purposes of this exercise we will treat all casual gains as having a 50% convective component. In a real project you would use values appropriate to the type of occupant, light or small power device.

The notes field allows space for recording assumptions and the intent of the data. Such notes help others decode the numbers within the schedules and are an essential part of QA. If the note mentions how many people or light fixtures this could be used to subsequently scale the data.

During model planning sketch the pattern of the various casual gains for each of the day types indicating the different periods and the magnitude of the gains. This information can then be used when inputting data as well as during model checking. Sketches save time. Try it for the data described above and compare this with Figure 1.9 for the reception and Figure 1.10 for the examination room.

This overview of how the building is used will be your reference material for much of the discussion of working practices in Chapter 5.

If you want to explore a variety of schedules for different building types, browse through the exemplar models and focus on how schedules are treated. Although not discussed in the *Cookbook* there are additional options for defining schedules of greater complexity. For example, there is a short timestep data facility which allows casual gains to be specified at each timestep.

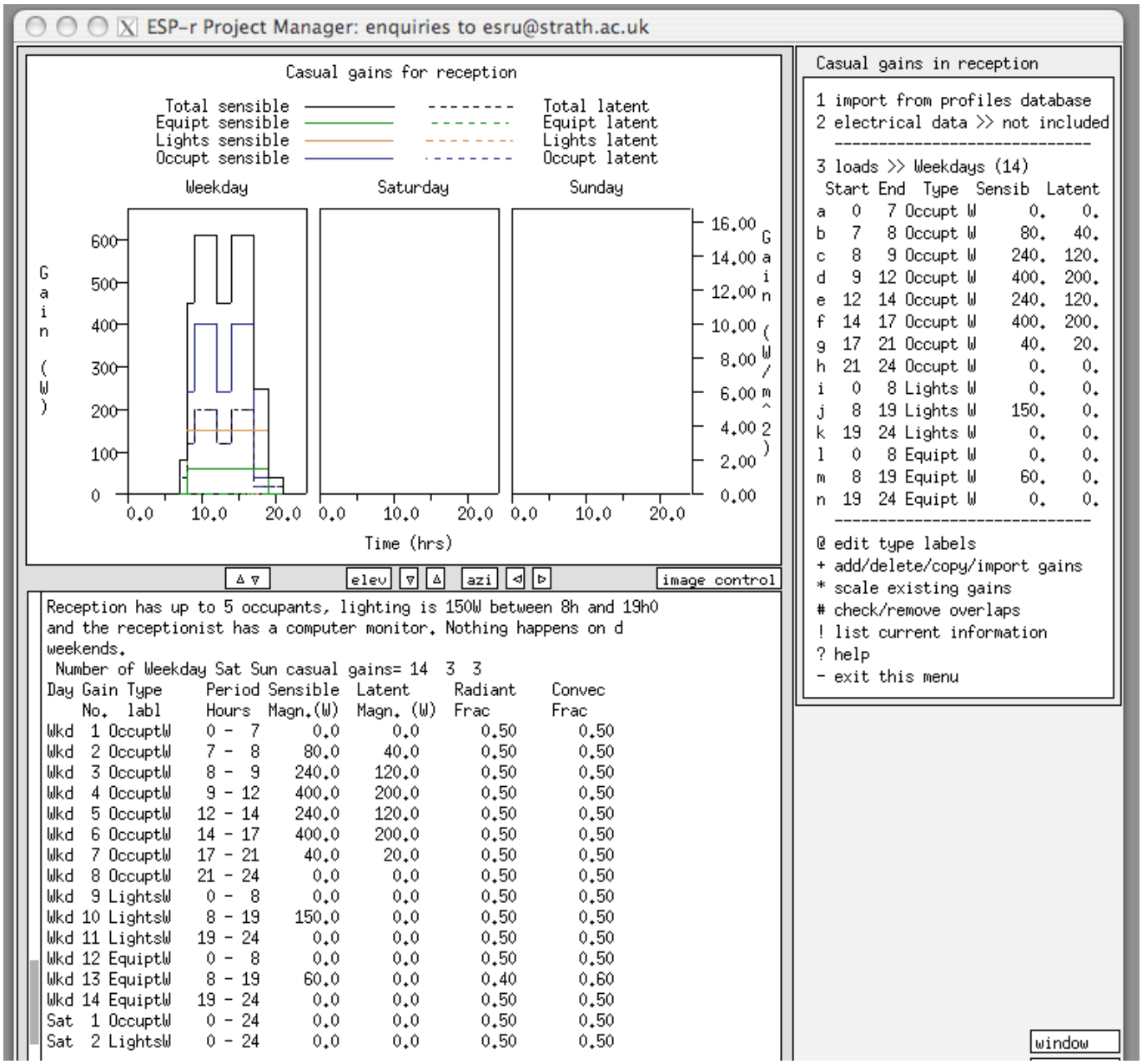


Figure 1.9 Profiles for reception.

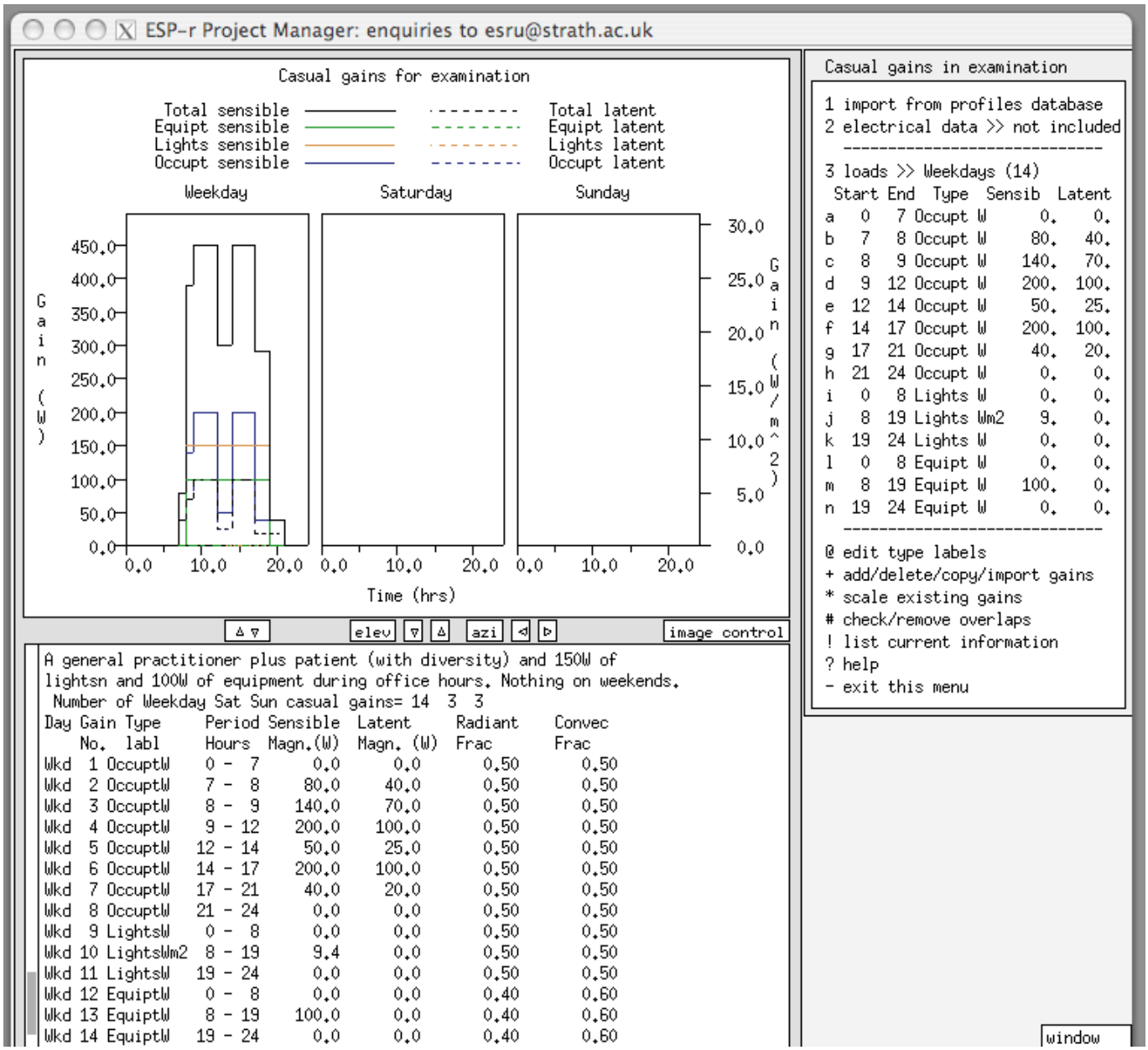


Figure 1.10 Profiles for examination.

1.7 Environmental controls

The *Cookbook* advocates a fast-tack strategy to establish:

- patterns of heating and cooling demand over time,
- the frequency of extreme conditions,
- the frequency of minimal demand,
- what might happen if heating or cooling failed,
- what might happen if heating or cooling was critically undersized,
- how often will the building work satisfactorily without mechanical intervention.

Such early indicators are valuable to other members of the design team. Deriving them may also result in well-founded opinions about demand side improvements and likely environmental control regimes.

Section 1.1 it did not include a specification for an environmental control system other than the set points to be maintained. Even if the brief had been specific it might not be well founded and would need to be evaluated.

Each simulation suite implements environmental controls via one or more arbitrary conventions:

- Ideal control laws which define what is sensed e.g. dry bulb air temperature, control logic that responds to the sensed condition and some form of actuation e.g. the injection of flux at some point in the model. Usually there are a limited number of parameters that can be set by the user and such controls tend to be applied to individual thermal zones.
- Ideal system descriptions which define a generally recognised pattern (e.g. VAV terminals with a perimeter trench heater), via high-level parameters which are then associated with a number of thermal zones in the model. Depending on the simulation tool there will be a different finite list to select from.
- Libraries of detailed system components e.g. fan coils and valves, which can be assembled by the user into a variety of environmental systems as required and linked with control components and logic.
- Templates which define an environmental control system from detailed components. The template expand a limited number of descriptive terms into scores, if not hundreds of components of a known topology, typically including control components and control logic. Templates often use a high level language to support the creation of component networks.

Software vendors have had mixed luck with each of these approaches. From the user perspective each has pros and cons.

- Ideal zone controls can mimic any number of real systems but present users with a mix of abstract terms e.g. radiant/convective splits and behaviour e.g. proportional/integral action rather than physical devices. Frustratingly, ideal zone controls often ignore the parasitic losses and electrical demands that many practitioners are interested in.
- Ideal systems, often only roughly approximate what the practitioner has in mind. Vendors have reacted

by adding more variants and/or providing additional input parameters. The chat lists for tools are full of practitioners confronted by the black art of tweaking an existing system to mimic a different design variant. Occasionally the mimicry takes on elements of a farce.

- Libraries of components are, in many ways a reaction to the constraints of pre-defined system lists. Opinionated practitioners are able to be specific and evaluate alternative designs and explore many more aspects of system performance. Unfortunately detailed descriptions are tedious to setup, difficult to calibrate and can approximate a black hole if they need debugging. Few interfaces or QA reports are able to communicate fully the attributes and relationships within a network of components. Somewhat to the surprise of software vendors, there are practitioners that lack the background, opinions or tenacity necessary to create systems from scratch.
- Which brings us to component networks created from templates. They offer the detailed performance characteristics of components with most of the tedium removed. Whereas the older ideal systems approach might have expanded a score of user inputs into a score of equations to be solved, the template approach can generate a network of scores, if not hundreds of components, and generate thousands of lines of description. Clearly the author of a template would have an advantage in understanding the resulting network composition and using it to support the design process. For others who have *any level of curiosity* about a newly created system the QA implications are substantial. Does the interface support understanding of what was created? If the practitioner needed to adapt or revise the parameters within components within such a network what methodology might they use to ensure this was done correctly?
  If *you think* your tool only offers detailed components, look closer to see if there are abstract components available. In the early stage of design they may be more than adequate.

ESP-r supports the following options for environmental controls within a model:

- Ideal zone controls (to be covered in this section)
- Ideal systems expressed as ideal zone controls with additional parameters to support post-processing of additional performance data e.g. flue losses, fan power. There is no interface to this facility and it is not supported on all computing platforms (a shameful state of affairs)
- Detailed system components, optionally in conjunction with mass flow network components and electrical power networks (to be covered in a later section)

At this point, many practitioners would, no doubt, feel compelled to jump into the details of their usual environmental control system. Rounding up the usual suspects is the antithesis of a strategic use of simulation. First, establish patterns of demand.

Second, explore options quickly by delaying specificity and detail in environmental control systems. Delivering information quicker and with less effort than our competitors *is a good business plan*.

Actually, many simulation tools make it difficult not to be specific in the early stages of a simulation project. Vendors sell Wizards that offer scores of predefined system templates which expand into networks of wondrous detail. Wow, so much from so little work (and it didn't crash so it must be a good design)! Vendors have less to sell with an abstract 'purchased air' option or ideal zone controllers.

So what approach to take? Here is a list of questions which might help identify whether a network of system components is suitable at the current stage of the design process:

- Do we have sufficient information to generate a network?
- What performance indicators of the network and/or components are we interested in?
- Can we explore broad-brush 'what-if' questions?
- Can we tweak component details during the detailed design phase?
- Are the physics within an idealised control or abstract component sufficient to explore a design issue? E.g. mimicking a radiant cooling system with 'purchased air' might be torturing the physics.
- What form of tool-generated documentation is available to support QA tasks?
- How does one validate a template-based system design? Is a systematic exploration of templates and components possible?
- Does the interface support detailed modifications of an initial template-based design?
- Does the interface support sufficient variants of an ideal control to allow a design team to pose relevant what-if questions?
- What support is available to move from an abstract representation to one with a higher resolution?

Since the focus is on learning about patterns of demand, an abstract description of the system is all that is required. ESP-r's ideal zone controls supports abstract descriptions of how heat or cooling is generated.

For purposes of this exercise an ideal zone control will characterize the response of a convective heating and cooling system to the client's set points. We do not know the capacity, so we will make an initial guess (say 4KW heating and 4KW cooling) and see how well that matches the demands.

Once these patterns are known our experience might suggest a sub-set of approaches. We can then explore different types of control logic and begin to be specific about the equipment that would be appropirate.

If the environmental controls have a schedule this should be recorded in much the same way as occupancy schedules. Indeed, there are often dependencies between occupancy patterns and environmental controls that

should be resolved at the planning stage.

If there are options for the type of system or the control actions which can be applied, ESP-r can accept alternate controllers which can be tested in subsequent assessments with little additional effort.

1.8 Model composition

The client has not specified what the building is to be made of. We are going to have to select placeholders from an existing database until such time as there is a clearer definition. Given the building type most professional practices will have available a number of likely constructions.

The following general types of constructions will be required for the building discussed in section 2.3:

- an exterior wall
- an internal lightweight partition
- double glazing for the windows
- a floor which includes some ground layers
- a ceiling for the sloped roof of the examination room which also acts as a roof
- a ceiling for the reception

One of our initial tasks will be to review the current contents of the construction and materials databases to locate existing entities which may be used as well as decide which ones can be adapated via copy and edit and which need to be added.

An example of the steps needed to do this are included in the *Cookbook Exercises* within Exercise 5.

This completes the planning phase and our next task is to create a model which matches the requirements set out in the planning stage.

2 Building a model

With planning complete we can take the client's specification and our sketches and notes and begin a new project. If you are using another simulation tool adapt your keystrokes and find equivalent facilities to create your model.

There are several exercises in the *Cookbook Exercises* volume which focus on this Chapter. Look at Exercise 2 as you start to create your model and complete Exercises 3-6 to ensure the databases are prepared with the entities you will need to build your model.

Interface interactions and typing are shown in `typewriter text.`

First select a folder for your model. Consider appropriate access privileges, how models will be archived and how models might be shared. A discussion of such issues can be found in the Install Appendix.

Give one of the following sets of command depending on the operating system your are using:

The following sequence will take you to your *home* folder and start up the ESP-r Project Manager:

```
cd
esp-r
```

Use Windows Explorer to select `C:\Esru\Models` or another folder that is not deeply nested and which has a minimum of spaces in the path. In the `C:\Esru\Models` folder there is an `esp-r.cmd` file which will startup the ESP-r project manager.

Figures 2.1 thru 2.4 illustrate (via the X11 interface) the steps we are going to take. Those using a GTK based interface will be asked the same questions, in the same order.

Our tasks is to create a new model so select menu option `Model Mangement -> create new`

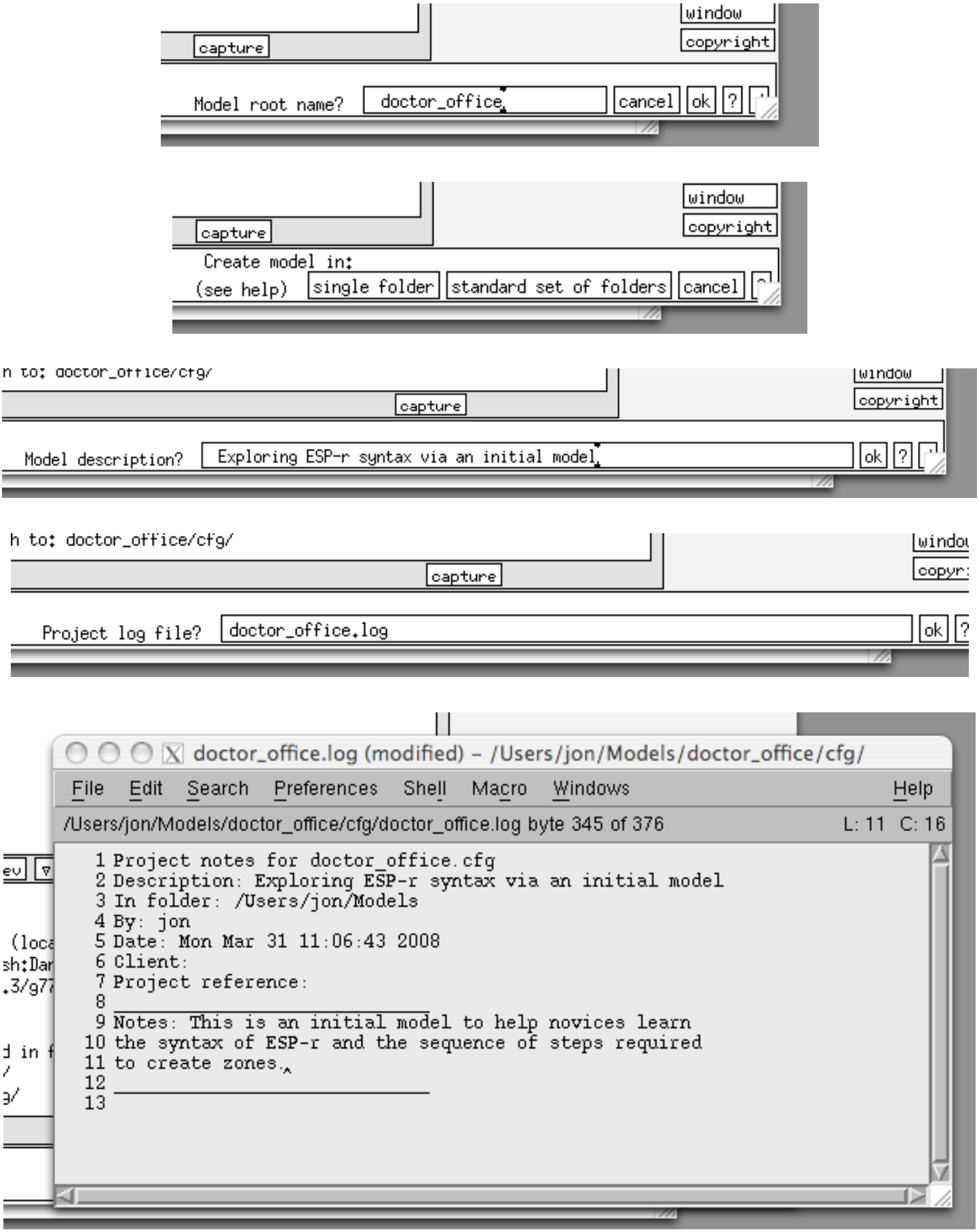


Figure 2.1 First steps in creating a new model.

```
Esp-r holds the model in a standard set of folders
and descriptive files.

Based on information you have supplied:

   GP_offices (project folder)
   GP_offices/cfg (system files)
   GP_offices/ctl (control files)
   GP_offices/zones (zone files)
   GP_offices/nets (networks)
   GP_offices/doc (reports and notes)
   GP_offices/temp (odds+ends)
   GP_offices/dbs (project databases)

will be created.
```

Figure 2.2: Folders to be created.

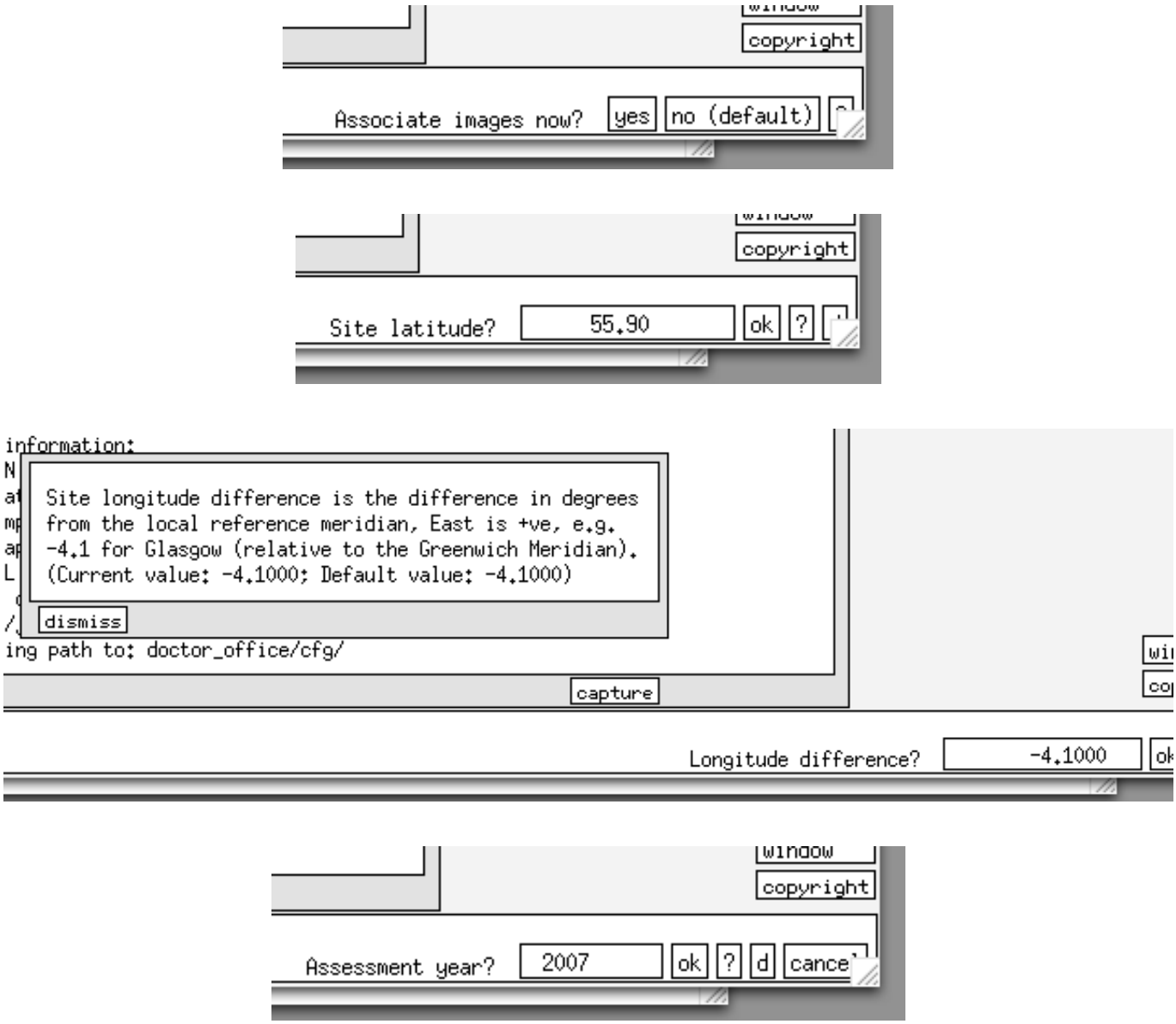


Figure 2.3: Further registration tasks.

A dialog will open at the bottom of the project manager which will ask you for a *root name* to use when creating the model and its folders (Figure 2.1). For this exercise lets use doctor_office. This root name will also appear in many of the model files so choose something which is short and clear.

You have the choice of placing your model files within a single folder or a standard set of folders. The single folder choice might be appropriate for a simple model. Since ESP-r separates model information into a number of files the standard approach is to use multiple folders to hold different types of information. For example, information about controls is held in a `ctl` folder and zone related information is held in a `zones` folder. For this exercise choose standard. You will be asked for a descriptive title for the model which is included in reports and above the wire-frame view.

There is a text log file associated with the model where you can keep track of who does what and when. You might also include in it a summary of the assumptions that you are making (in case someone asks). QA tasks are **so much easier** if you take a few moments documenting your model.

Notice in Figure 2.3 that the `Longitude difference?` dialog shows a pop-up help message. All dialogues and menus have contextual help. When in doubt use the `?` button.

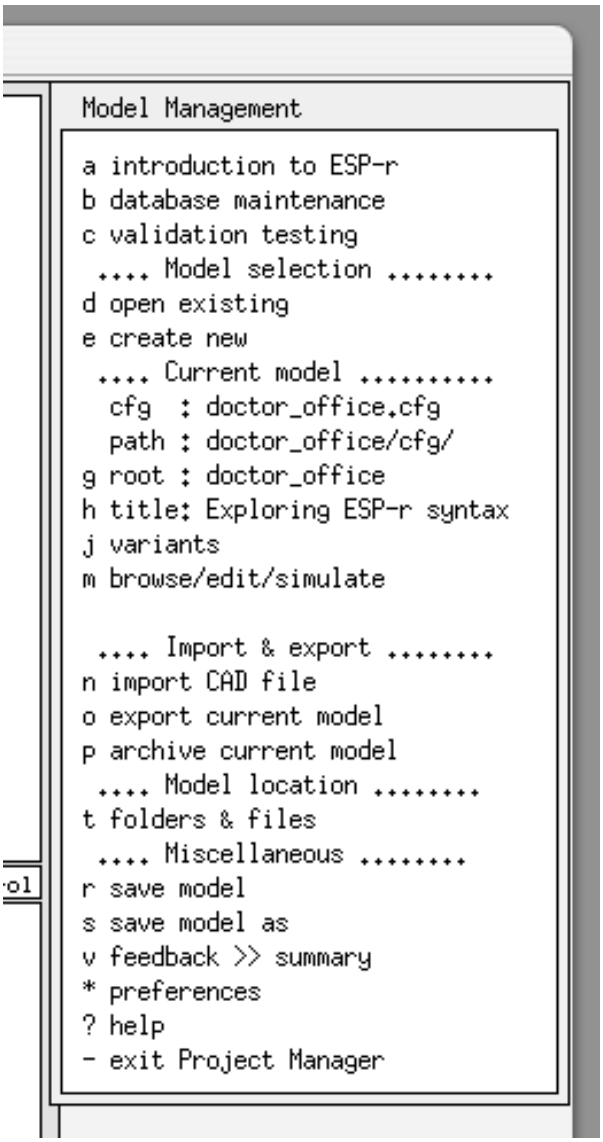


Figure 2.4: Model management menu.

2.1 Review of climate patterns and databases

At this point we have registered a new simulation project (the terms project and model are often used interchangeably in ESP-r). There are a number of tasks that we want to complete before we begin to define the form and composition of the general practitioners office. This section and Exercise 3 in the *Cookbook Exercises* are concerned with the following tasks:

- Find a climate file and typical climate periods for our assessments.
- Review construction and materials databases.
- Select or create places holders for constructions and materials.

The *Cookbook* advocates the use of short climate sequences for model calibration and focused explorations. For example, a Monday morning startup after a cold weekend can tell us much about the characteristics of a building. Do peak summer demands coincide with the hottest day or is it a function of several hot days in sequence? There is no point in using an annual assessment to address such issues and, more importantly, great advantage to front-loading simulation tasks so that models are calibrated as soon as possible.

The following discussion includes climate data search techniques for identifying an appropriate week in winter with a cold weekend and a summer week with a sequence of warm days.

To work with climate databases, select the menu option `Model Management -> Database Maintenance` and in the options shown in Figure 2.5 select the `annual climate` option.

Once a database is selected then there are a (mostly) common set of tasks which are available (see Figure 2.6). Some databases include functions to convert between binary and ASCII versions. Databases which require frequent random access typically have a binary form. ASCII versions are useful for transport between computers.

For climate databases there are also options to convert EnergyPlus EPW files and Korean MET office files to ESP-r format file.

Assuming ESP-r was installed correctly, you will now be presented with a list of known climate files (see Figure 2.7). Adding additional sets is a separate topic.

For purposes of this exercise we want to select an existing climate data file. Choose the Birmingham IWEC climate, look at the summary in the text feedback area and confirm the selection. The ESP-r climate module (**clm**) will start, and your first task is to confirm the climate file name in the intial dialogue.

The ESP-r climate module (**clm**) provides facilities to explore climate data sets via graphs, statistics and pattern analysis facilities (see Figure 2.8). Our initial task is to use these facilities to better understand the climate patterns in Birmingham and identify useful assessment periods.

```
Database Maintenance

  model path: /Users/jon/cellular_shd/cfg/
  -------------------------------
a annual climate          : /usr/esru/esp-r/climate/clm67
b multi-year climate      : None
c pressure distributions  : /usr/esru/esp-r/databases/pressc.db1
d materials               : ../dbs/cellular.materials.a
e constructions           : ../dbs/cellular.constrdb
f plant components        : /usr/esru/esp-r/databases/plantc.db1
g event profiles          : /usr/esru/esp-r/databases/profiles.db1
h optical properties      : /usr/esru/esp-r/databases/optics.db2
i mould & mycotoxins      : None
  -------------------------------
? help
- exit this menu
```

Figure 2.5 List of ESP-r databases.

```
Materials db:

a browse/edit
b select another
c create new
d copy standard
e binary >> ascii export
f ascii >> binary import
g ascii >> memory import
  ----------------------
* default is option `a`
? help
- exit this menu
```

```
Constructions db

a browse or edit db
b select another db
c create a new db
d copy the default db
  ----------------------
* default is option `a`
? help
- exit this menu
```

```
Climate db

a browse/edit db
b select another db
c create new db
d copy default db
e binary >> ascii export
f ascii >> binary import
g EPW >> binary import
  ----------------------
```

Figure 2.6: Typical options for databases.

```
Climate sets

ESRU standard climates
-------------------------
a Default UK clm Climate
b ALBUQUERQUE NM USA iwec 723650
c BALTIMORE MD USA iwec 724060
d Belfast IWEC WMO 039170
e Berlin Germany IWEC 103840
f Birmingham IWEC WMO 035340
g BOISE ID USA iwec 726810
h Brussels Belgium IWEC WMO 0645
i BURLINGTON VT USA iwec 726170
j Calgary Intl AB CAN
k CHICAGO IL USA iwec 725300
l Copenhagen Denmark IWEC 061880
m DIJON - FRA (47.27N 005.08E)
n Dublin Ireland IWEC WMO 039690
o Dundee 1980
p DULUTH MN USA iwec 727450
q EL_PASO TX USA iwec 722700
r Jersey CH ISL IWEC WMO 038950
s FAIRBANKS AK USA 702610
t Finningley IWEC WMO 033600
u Firenze-Peretola - ITA
v Gatwick IWEC WMO 037760
w GENEVA - CHE
x GUANGZHOU - CHN
0 page:  1 of  3 ------
< user defined climate db
* new climate db (all zeros)
? help
- exit
```

Figure 2.7: Available climate sets.

There are a number of options under synoptic analysis (Figure 2.9) which are useful for finding climate patterns with which to test our building.

To generate a statistical report as in Figure 2.10, first select the climate data you wish to analyze e.g. `dry bulb temperature` and then the type of analysis e.g. `maximum & minimum` and then the reporting frequency e.g. `day/week/month.` Near the bottom of the is an option `find typical weeks.`

This facility works as follows:

- the average & total heating and cooling degree days (HDD & CDD) and solar radiation are determined for each season,
- for each week, average HDD & CDD and solar data is found and compared with the seasonal values and the week with the least deviation (using user supplied weighting factors) is reported. For this climate and with the default heating and cooling base temperatures the best-fit weeks start on 27 Feb, 10 April, 19 June, 5 Oct and 4 Dec. Write these dates down and then go review these periods by graphing and/or gathering statistics about them.
- The climate module provides several ways of looking at the data so see which 'view' tells you the most!

The provision of different views of the climate data can assist in locating patterns within the climate data and answering different questions that clients might pose.

Time spent exploring this module can provide critical clues as to patterns within a climate that may be used in the design process.

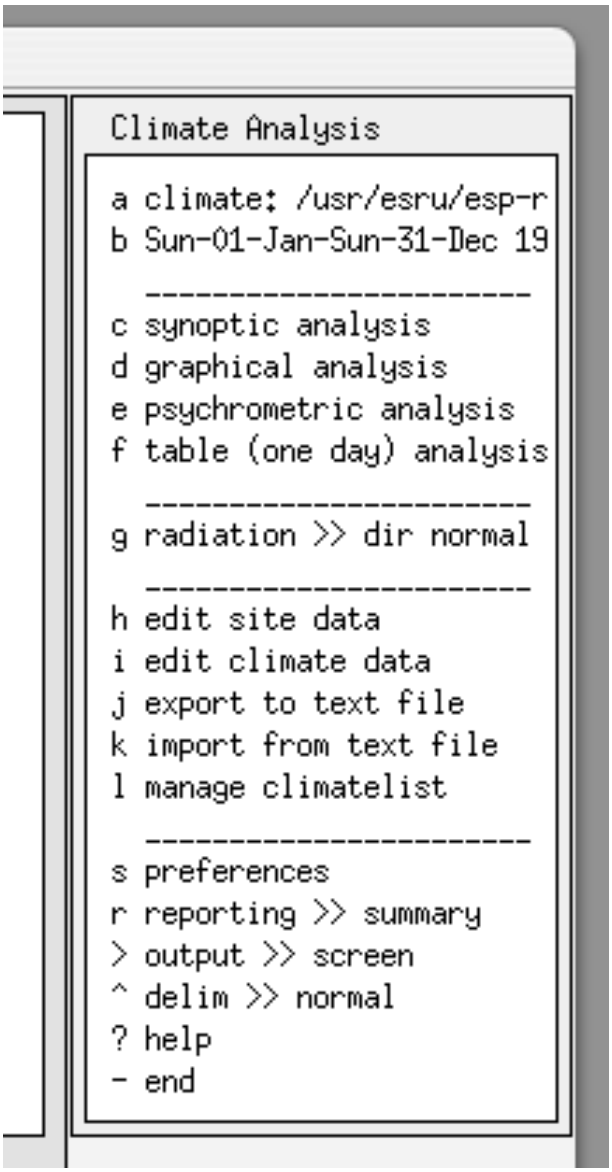


Figure 2.8: Clm module opening menu.

An example is the graph of temperatures over the year (Figure 2.11) with the current seasons indicated across the top of the graph. There are a number of times when it is below freezing, but the graph indicates that these tend to be brief. This might support the use of brief performance assessments for winter heating demands and capacity. It also indicates scope for testing whether a design might be optimised to cope with brief rather than extended cold periods.

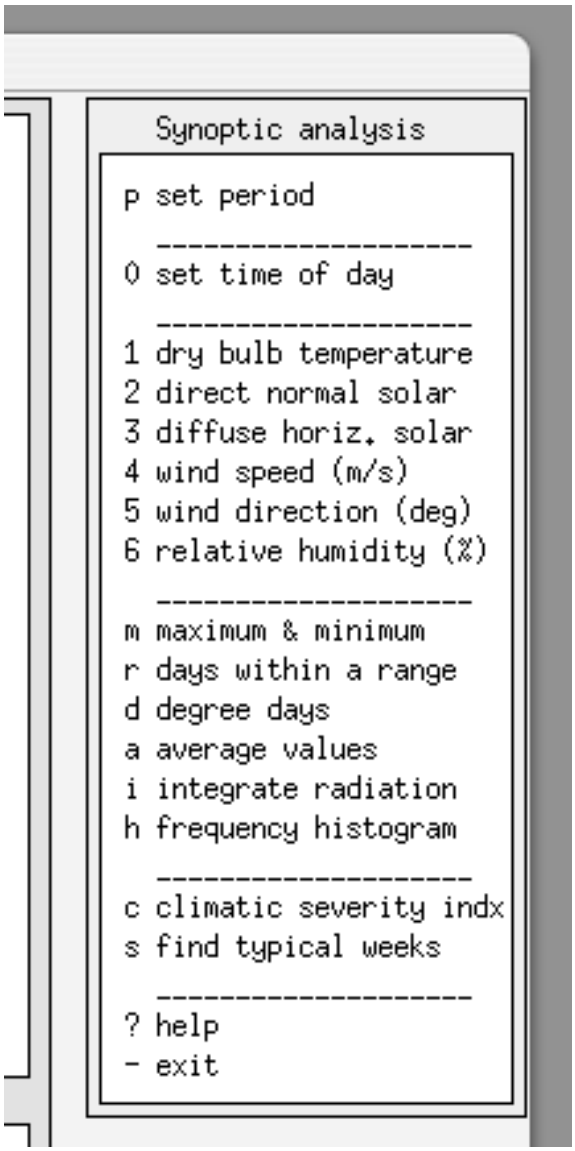


Figure 2.9: Synoptic analysis.

Another example is Figure 2.12 where the psychrometrics of the outside air have been plotted over the whole year for the same location.

Most companies who regularly deploy simulation will have evolved procedures for selecting climate data for specific design assessments. The acquisition of climate data is covered in a later chapter.

Dry bulb temp. deg.C

Week	Minimum	Time	Maximum	Time	Mean
Wk of Wed 1 Jan	-15.0	@24h00 Tue 7	6.8	@ 4h00 Wed 1	-5.2
Wk of Wed 8 Jan	-23.9	@ 7h00 Sun 12	2.8	@17h00 Fri 10	-15.3
Wk of Wed 15 Jan	-23.3	@ 1h00 Thu 16	-1.1	@14h00 Mon 20	-9.6
Wk of Wed 22 Jan	-25.0	@ 5h00 Mon 27	-5.0	@ 4h00 Thu 23	-14.4
Wk of Wed 29 Jan	-22.8	@ 8h00 Wed 29	-10.0	@18h00 Thu 30	-15.0
Wk of Wed 5 Feb	-15.8	@ 1h00 Thu 6	-3.6	@16h00 Fri 7	-10.0
Wk of Wed 12 Feb	-16.9	@ 7h00 Mon 17	-1.3	@16h00 Fri 14	-8.1
Wk of Wed 19 Feb	-14.4	@24h00 Tue 25	4.7	@14h00 Thu 20	-3.9
Wk of Wed 26 Feb	-21.1	@ 7h00 Wed 26	5.6	@14h00 Sun 2	-5.5
Wk of Wed 5 Mar	-12.2	@ 5h00 Thu 6	9.4	@10h00 Wed 5	-3.6
Wk of Wed 12 Mar	-10.6	@ 7h00 Sun 16	5.6	@23h00 Fri 14	-3.0
Wk of Wed 19 Mar	-11.7	@ 6h00 Wed 19	11.1	@17h00 Mon 24	-0.4
All period	-25.0	@ 5h00 Mon 27 Jan	11.1	@17h00 Mon 24 Mar	-7.6

Figure 2.10: Weekly statistics.

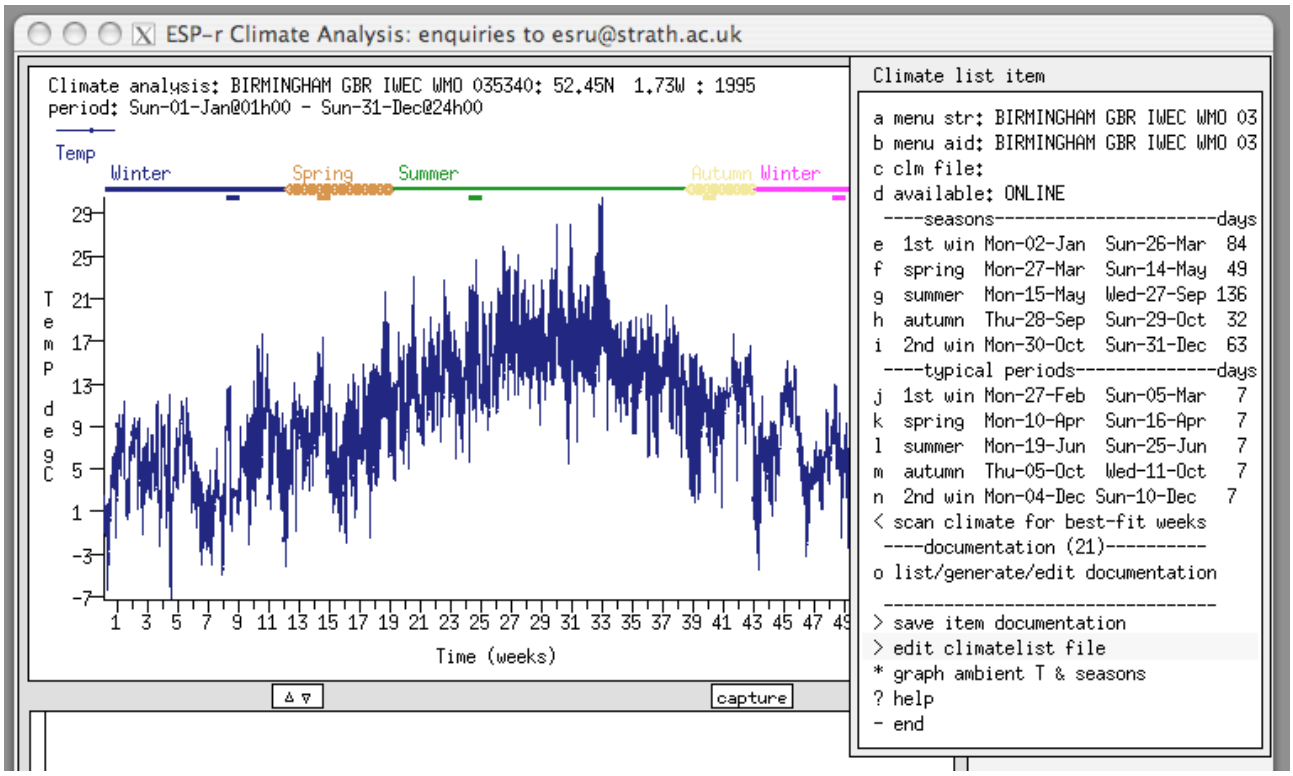


Figure 2.11: Annual plot of temperatures.

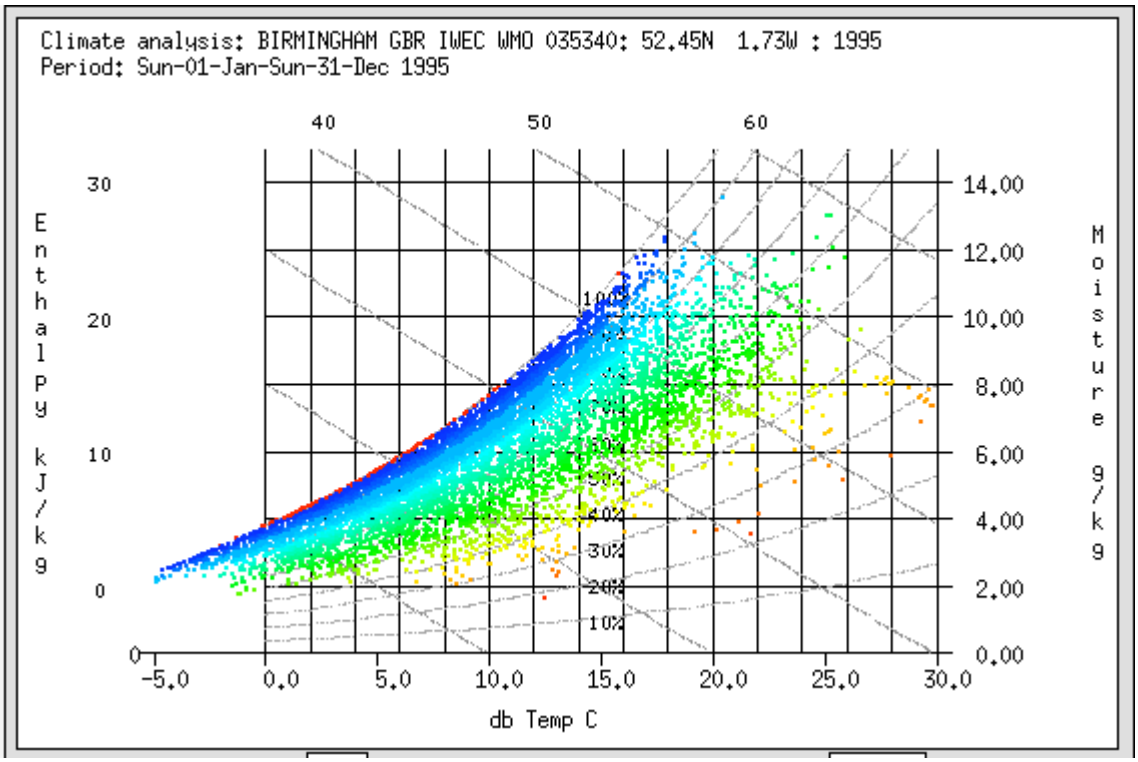


Figure 2.12: Annual psychrometrics.

2.2 Locating constructions for our model

Our next task is to do a quick review of available constructions so that we will be able to attribute our model as we create it. You will want to look at Exercise 4 which focuses on a review of materials which may be needed and the steps required to take an existing material and create a variant which is appropriate for the current project. Exercise 5 will give you practice at the review of constructions as well as in adapting existing constructions and creating new constructions. The exercise uses the standard ESP-r constructions database. If you are working with a different set of databases the process is the same but you will have to adapt the details to fit.

It is possible to configure ESP-r to load an initial set of databases that are appropriate for a given region and building type (see Install Appendix). Typically material and construction databases are managed at a corporate level but ESP-r also allows model specific databases and possible working practices are also discussed.

So, return to `Model Management -> database maintenance -> constructions` select the current (e.g. standard) database and note the list of constructions (see Figure 2.13) and details (see Figure 2.14) of those you might associate with this new model.

One tactic, if you don' t find what you are looking for, is to make a project copy of one of the standard databases and then add in place-holder constructions. A place-holder is a named construction which either does not have a full set of thermophysical properties or uses an approximation. This can be associated with the surfaces in the model and later, when the actual information is available it can be updated

and all of the surface which use it will take on the revised properties.

```
Constuctions database

a extern_wall   OPAQUE
b insul_mtl_p   OPAQUE
c intern_wall   OPAQUE
d partition     OPAQUE
e door          OPAQUE
f int_doors     OPAQUE
g mass_part     OPAQUE
h int_part      OPAQUE
i d_glz         DCF7671_06nb
j dbl_glz       DCF7671_06nb
k roof_1        OPAQUE
l roof          OPAQUE
m roof_2        OPAQUE
n grnd_floor    OPAQUE
o floor_1       OPAQUE
p entry_floor   OPAQUE
q susp_ceil     OPAQUE
r susp_floor    OPAQUE
s susp_flr_re   OPAQUE
t ceiling       OPAQUE
u ceiling_rev   OPAQUE
v gyp_blk_ptn   OPAQUE
w gyp_gyp_ptn   OPAQUE
x switched_gl   DCF7671_6omb
0 ---Page:  1 of  2 ---
# view g-value
1 add/delete/copy/invert
! list database contents
> save database
? help
- exit this menu
```

Figure 2.13 List of available constructions in database.

```
Construction editing

a Construction: dbl_glz
b General type: Transparent
c Optical properties: DCF7671_06nb
  No of layers: 3 (   24.0mm thick)
d Layers are: SYMMETRIC

  ----------------------------
 Layer|Prim|Thick| Description
      |db  |(mm) | of material
i   1  242    6.0  plate glass
j   2    0   12.0  air  0.17 0.17 0.17
k   3  242    6.0  plate glass
 ISO 6946 U h/u/d 2.811 3.069 2.527
  ----------------------------
! add or delete a layer
? help
- exit this menu
```

Figure 2.14 Construction data.

When you complete Exercise 5 add to your project notes the names of the existing and new constructions which you wish to associate with your model. This saves time and reduces the chance of error!

2.3 Zone composition tactics

Before we begin defining our zones lets review tactics. For purposes of this exercise we are going to use the in-built CAD facilities. We are going to use these facilities tactically so as to minimize the number of keystrokes and avoid errors. Later on you can adapt the techniques for even greater efficiency. Here are the tactics:

- plan for maximum re-use of existing information
- use information on your planning sketches and notes
- take opportunities to embed documentation in the model
- give entities meaningful names and use attribution early-on in the processing
- learn the tool well enough to use in-built facilities to copy, edit and transform

The order we define a model can allow us to build new zones from portions of adjacent zones. In this exercise if we begin with the reception we can make use of this information when creating the examination room.

If we take the information from our planning sketches rather than improvising or using a calculator, we will make fewer errors, we will be less likely to loose track of where we are when the phone rings and we will be less likely to find we need two more surfaces than the interface allows.

ESP-r has numerous places where you can document your assumptions. Of course you would never want to use such facilities because you never loose scraps of paper and you always remember the assumptions you made about that model four months ago and your clients solicitor will never ask you to prove you followed procedures or used the correct values for that computer lab.

A surface in a simulation model is not just a polygon, it has a name, it is made of something, it has specific boundary conditions and it must conform to the (virtual) laws of physics. QA gets a lot easier if something that looks like a door in a wireframe image, is named door and is composed of a door type of construction. The tactic is for these attributes to reinforce each other so it is easy to notice if we get something wrong and to be able to easily focus on the correct portion of our model.

Defining the reception

Returning to the `Model Management` menu find the `*preferences` menu option and set ESP-r to use the most recent geometry file format as in Figures 2.4 and 2.15. The option labelled `upgrade files -> scan and update all zones` sets the preference.

Next we want to traverse the menu structure to focus on the geometry of a new zone as shown in Figures 2.16 - 2.18. Select `Model Magement -> browse/edit/simulate -> composition -> geometry & attributions.` Since there are no existing zones you will be asked whether you want to `input dimensions` or `load existing (ESP-r), load existing (CAD)` or `cancel.` Choose to `input dimensions,` name the zone

e.g. `reception` is a good choice.

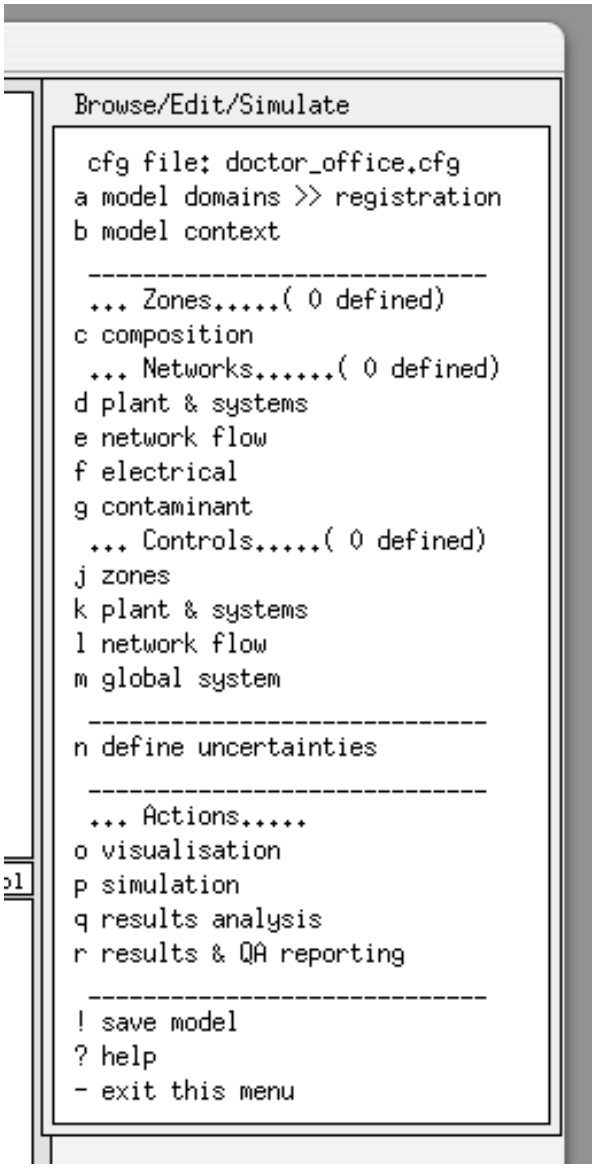


Figure 2.15: Browse/Edit prior to adding zones.

Next you will be asked to describe this zone - so lets paraphrase of the clients definition.

The next dialog asks whether we want to start with an `rectangular plan`, `polygon plan`, `general 3D`, `bitmap`. For purposes of this exercise select `polygon plan` and find your sketch which corresponds to Figures 1.1 and 1.8.

This type of input requires the height (in real space) of the floor (0.0m) and ceiling (3.0m).

How many walls? Figure 2.17 indicates an additional point at X=4.0 and Y=7.0 where there is a break in the wall. To the West of this break the wall is a partition which faces another portion of the building which we are not going to bother to define. To the East of this point the wall faces the outside. We need to define two walls along the north side of the reception and so the total number of walls that we are going to extrude from the floor plan is 7.

The arrows shown in Figure 2.17 indicate the ordering of the walls. Remember this rule: when extruding a floor plan proceed anticlockwise. Write down the critical co-ordinates first and don't answer the phone.

In this case we will start with 0.0,4.0 and then 4.0,4.0 etc. around to 0.0,7.0 (we need not repeat the initial point).

When the walls coordinates have been defined you have the option to accept them. If you are in any way worried about having made a mistake, say no and you will have a chance to check and edit the data.

The `Rotation angle?` dialogue allows you to define co-ordinates in a cardinal orientation and then rotate to reflect site conditions. You can also perform transforms and rotations later. To skip the rotation, leave the rotation angle at zero in Figure 2.18.

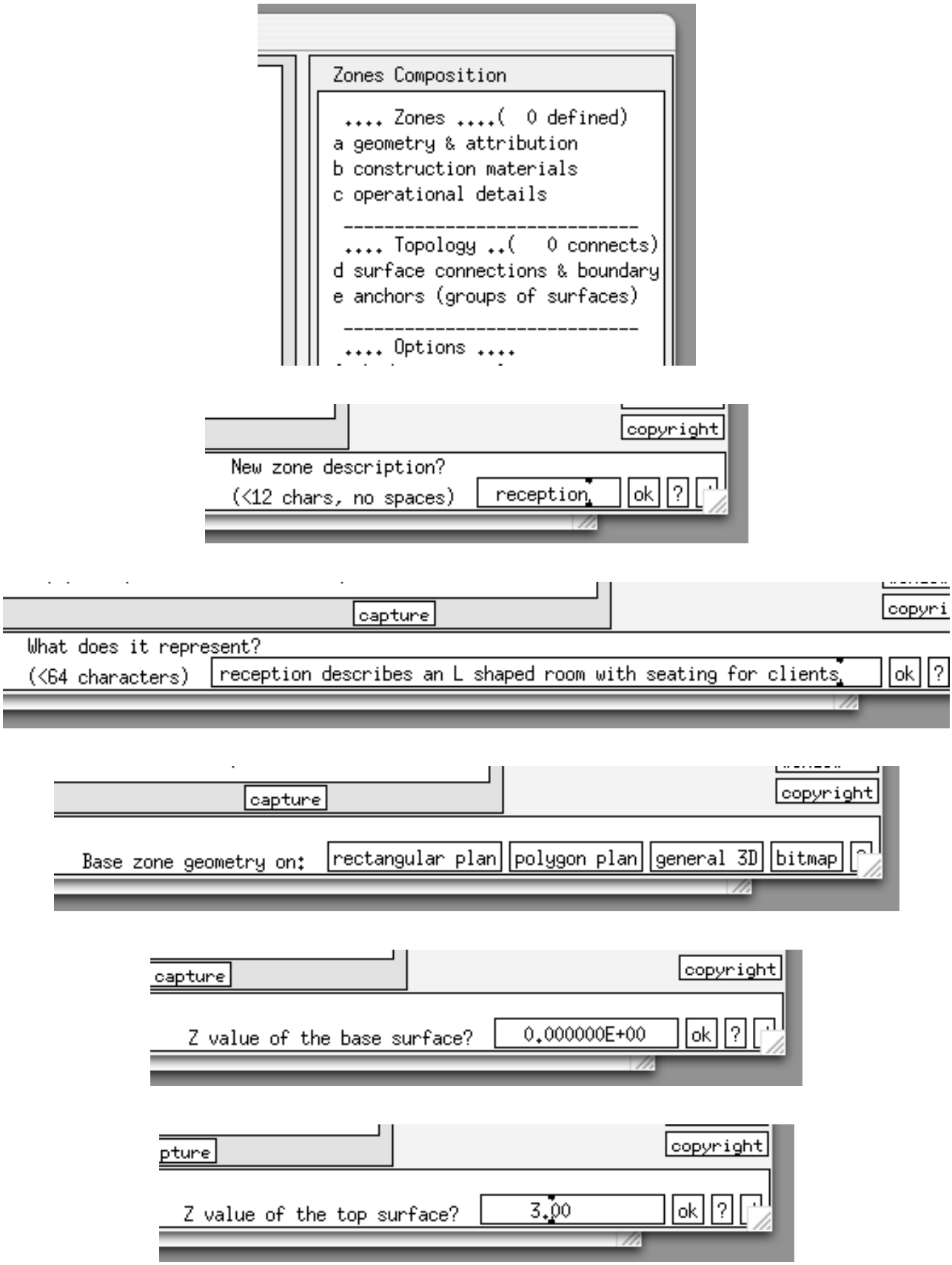


Figure 2.16: Zone name, description, shape.

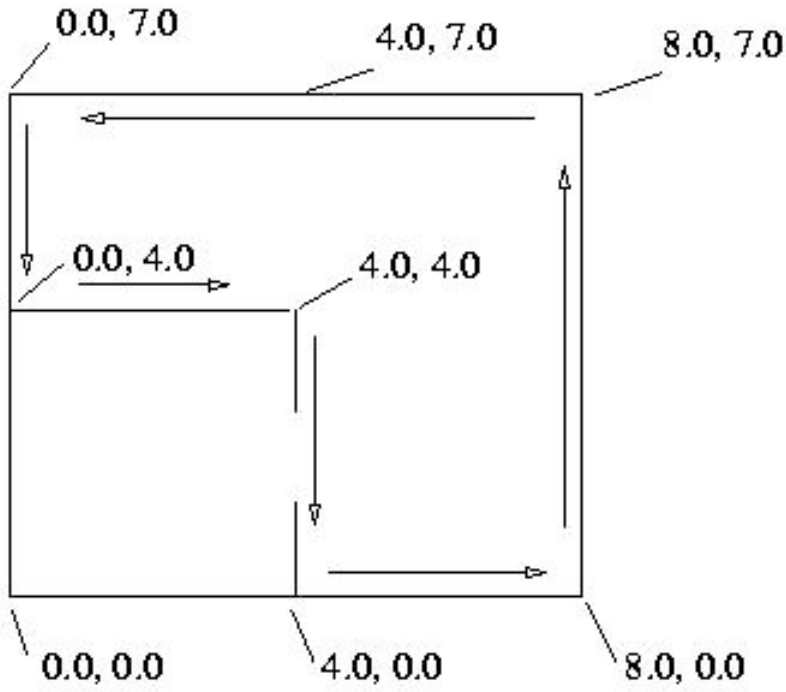


Figure 2.17: List of plan coordinates.

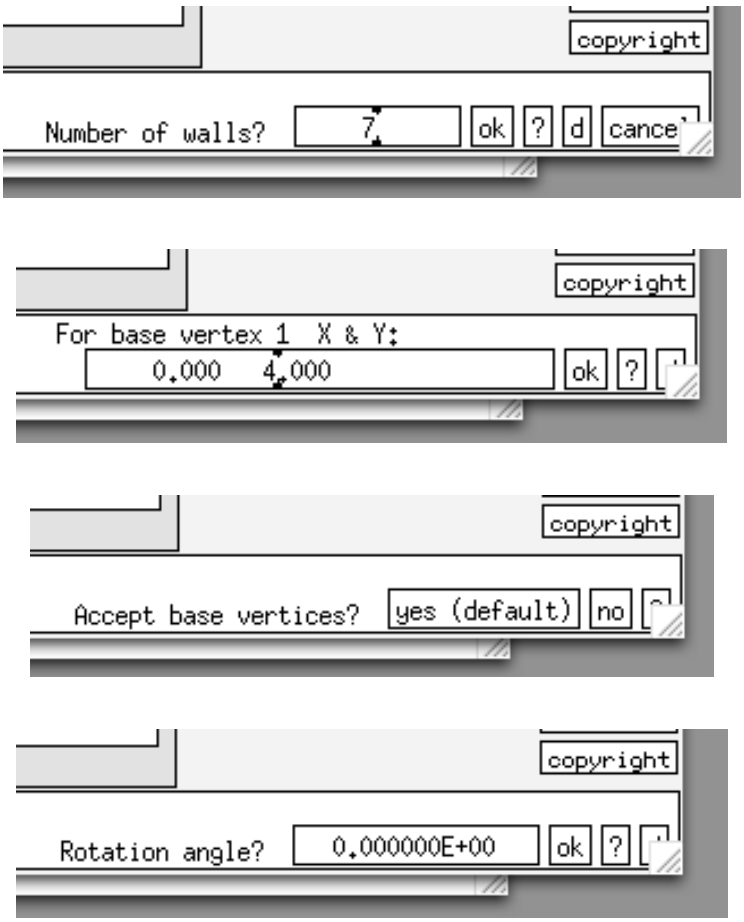


Figure 2.18: Steps required for extrusion.

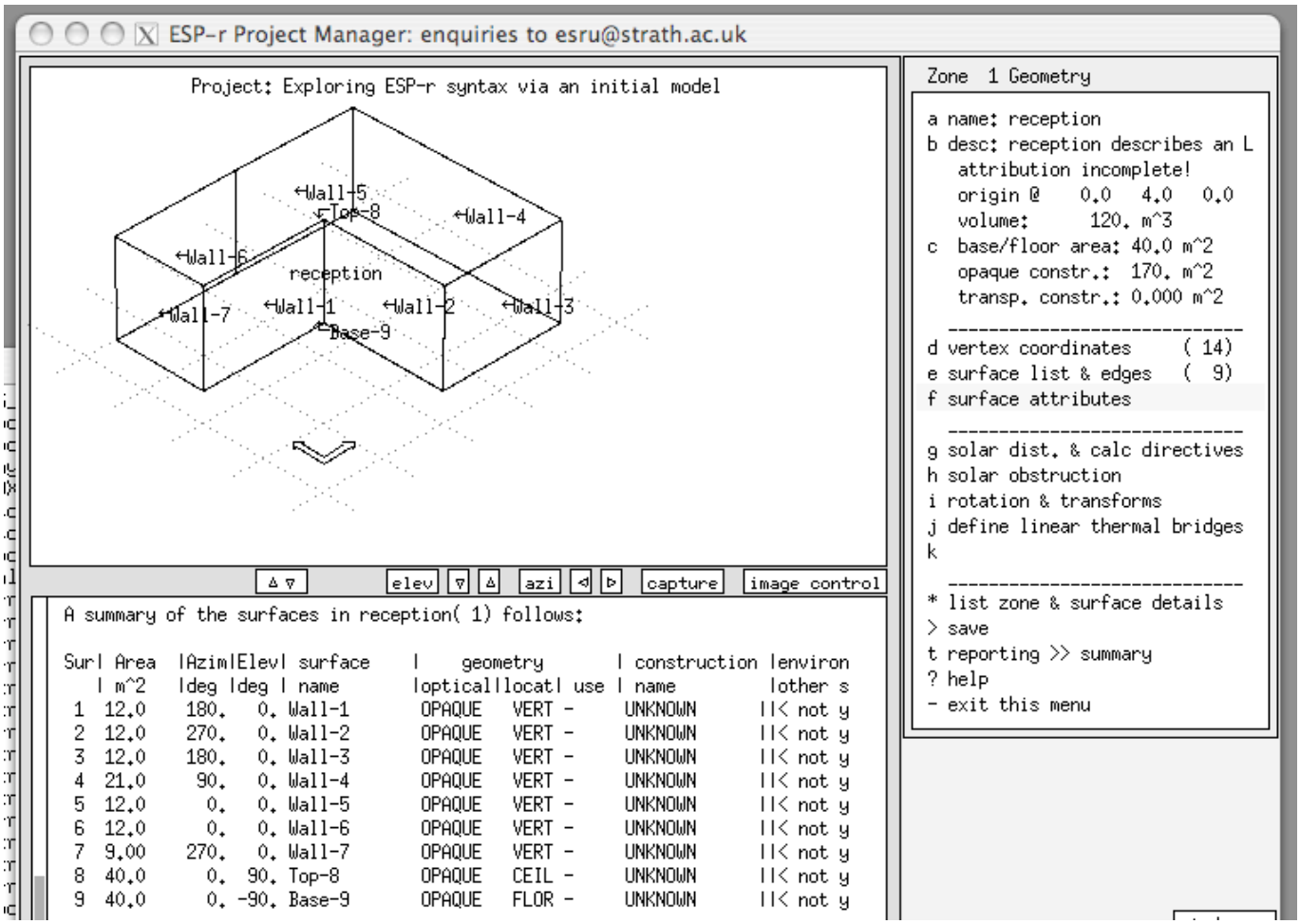


Figure 2.19: And here is what it look like!

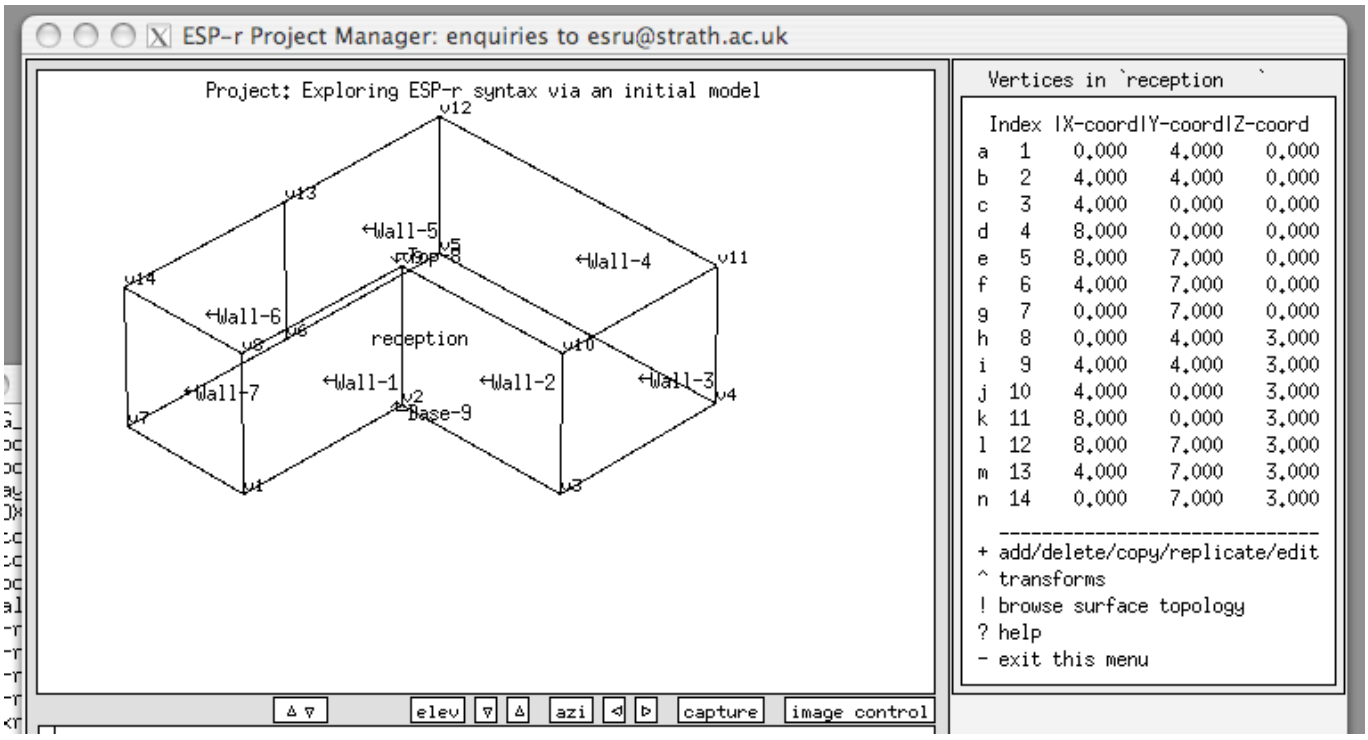


Figure 2.20: Initial list of co-ordinates.

```
Surface topology of reception

  enclosure: properly bounded

  -------------------------------
  Surface   |No. |Verts (anti-clk
  name      |vert|from outside)
a Wall-1        4  1  2  9  8
b Wall-2        4  2  3 10  9
c Wall-3        4  3  4 11 10
d Wall-4        4  4  5 12 11
e Wall-5        4  5  6 13 12
f Wall-6        4  6  7 14 13
g Wall-7        4  7  1  8 14
h Top-8         7  8  9 10 11 12..
i Base-9        7  1  7  6  5  4..
  -------------------------------
+ add/insert/copy/extrude_from
* delete a surface
> transforms
! browse surface-vertex topology
@ check surface-vertex topology
? help
- exit this menu
```

Figure 2.21: Initial list of surface edges.

Once you accept the points the project manager display will update to what is shown in Figures 2.19 and 2.20 with a listing of the vertex associations in Figure 2.21 are several things to notice about the graphic feedback and the command options.

- Items in the menu which start with a character can be selected and those that start with a blank in the first column are reporting derived values (which will be updated as you modify the model).
- Under the graphic feedback area (of the X11 interface) are a number of buttons for altering the viewpoint, altering the size of the graphic feedback area and controlling the image. The latter allows you to turn on and off names and vertices and the site origin. Spend a few moments exploring this. The GTK+ interface includes a proforma for adjusting viewing parameters (see Version Appendix)
- The text feedback shown in Figure 2.19 includes a synopsis of the geometry of the zone and a list of the attributes and derived values for each of the surfaces. This report is designed to *complement* the wire-frame image and you will notice that the word UNKNOWN shows up many times so there is still some work to be done!
- You will notice that the surfaces have been given names like Wall-1 and Base-9. These are unambiguous without being particularly helpful.

Doors and windows

In ESP-r, windows and doors should be included as surfaces in a model if they are *thermally important*. A tactical approach also includes windows and doors if they make it easier for *others to understand* the model. The door probably falls into the latter category - we save time because we will not have to explain why we have omitted it.

Essentially a surface is a window or door if you decide it is and give it a name that reminds you of this decision. This is somewhat different than other simulation tools so it is worth reviewing the rules:

- any surface in a model can be glazing or a door with the exception that a transparent surface cannot be used for a back-to-back connection representing mass *within a zone*

- their shapes are constrained only by normal polygon rules
- if glazing has no frame there is no requirement that it be a *child* of another surface
- doors and glazing can be bounded by more than one surface (e.g. glazing could have a frame on the left and top and right but adjoin a spandrel panel on the bottom edge)
- the base of a door (or glazing) can be at floor level or it can have a raised sill
- door and window composition can be any valid construction (e.g. doors can be transparent or opaque and windows can be opaque even though that would tend to confuse others)
- if you want to represent the *adjacent-to-frame* portion of glazing differently from the *centre glass* then you can either adjust the thermophysical and optical properties or create two separate glazing surfaces

In ESP-r, solar radiation will pass through any surface which has an optical attribute set. This is the case for surfaces facing the outside as well as surfaces which are acting as partitions with another zone. A construction can use glass as a material and will consider the glass as opaque if it is matched to an OPAQUE optical property.

Thermophysically a door is treated like any other surface in ESP-r. Surfaces with an optical property will transmit and absorb solar radiation but are otherwise treated as any other surface. Other simulation tools may treat doors and windows as simplified entities e.g. a solar heat gain fraction might be used rather than explicitly treating the radiation and convention from glazed surfaces.

You have several choices in the treatment of window frames. You can explicitly represent frames as one or more surfaces in the zone. A frame may wrap around the glazing or you can take an abstract approach and lump all of the framing facing one direction in a room into a single surface.

You also have several choices for glazing. If the accurate positioning is important then create zones with explicit representations. If you only require that the area and orientation of the glazing is important then you could choose an abstract approach and lump all of the glazing of one type and orientation into a single surface.

One issue which you may wish to consider is the relative size of surfaces in a zone and the interactions between surfaces:

- if a large pane of glass is used in a room with small surfaces (e.g. explicit blinds) then you may wish to subdivide the glass
- if you are calculating view-factors then small dimensions may confuse these calculations - a frame 20mm wide around a large glazing or door surface may not get enough grid points for an accurate calculation.
- if you are interested in local comfort then you should consider increasing the geometric resolution in the portion of the zone where comfort is being assessed.

- if the impact of solar distribution is important (e.g. a passive solar design) then you may want to subdivide the floor and the major thermal mass to account for the patches of sunlight as well as using explicit representations of glazing.

In ESP-r, air passes between rooms ONLY if you define an air flow schedule or create an air flow network. It is optional to include a door, grill or window surface at the air movement entity location. Such visual clues may help to clarify a model.

If your ESP-r model is going to be exported to an application which has a different set of rules for windows and doors then also take those rules into account (if possible). For example, Energy Plus requires glazing to be a child surface of an opaque parent surface so ensure that glazing in ESP-r models are also represented this way. Test various approaches to find which works best.

Our next tasks (shown in Figures 2.22 - 2.27) is to insert a window into Wall-3 and Wall-5 and a door into Wall-2. The interface provides a way to make a rectangular 'hole' in an existing surface and place a new surface into that hole. The original (parent) surface wraps around all sides of the child surface.

In Figure 2.22 there are a number of options for creating new surface. You will use these options frequently so it is worth the time to explore them to identify which of the variants might be used in different situations. The interface also provides an option to insert a door into a surface. It does this by wrapping the existing surface around the sides and top of the door.

In both cases you will be asked to provide an offset and a width and height of the rectangle to be inserted. The offset is from the lower left corner of the existing surface (looking from the outside). These inbuilt facilities support the insertion of rectangular child surface. There is no specific requirement in ESP-r that doors and windows must be rectangular. If the design includes an arched opening or a round window you can do this (but the surface will have to be made of a number of straight segments).

According to Figure 1.1, the window in Wall-3 starts 2.0m above the floor and is centered in the wall. So the offset X is 0.5m and the offset Z is 2.0m and the window size is 3.0m x 0.75m (see Figure 2.24). The same offsets and size apply to the window to be inserted into Wall-5. The door is 1.5m from the edge of Wall-2 and is 1.0m wide (see Figure 2.26). No height is given in the specification so lets assume 2.1m.

Beginning with the window in Wall-3, select `surface list and edges -> add/insert/copy/extrude_from` and then choose `inserted into a surface -> within surface.` Notice there are also options to insert a surface as `percentage of parent surface.`

```
new surface options

a made from existing vertices
b made from existing vertices (mouse)
c inserted into a surface
d copy surface(s) in this zone
e copy surface(s) from another zone
f vertical rectangle (origin&azim)
g horizontal rectangle (origin&rot)
h extrude sides/top from base surface
i vertical rect mass (origin&azim)
j horizontal rect mass (origin&azim)
  -------------------------
* default is option `c`
? help
- exit this menu
```

Figure 2.22: Options for creating surfaces.

Provide the dimensions from the paragraph above and the proposed location of the window will be shown on the wireframe. If it is **not ok** you can re-enter the data, otherwise a new surface will be added and the existing Wall-3 which started out with 4 edges becomes a 10 edge surface which wraps around the new surface. You will be asked to name the new surface (south_glz or something similar) and pick a construction for it (there is an entry dbl_glz in the constructions database).

A recent addition to ESP-r is the concept that surfaces have a *usage attribute* (Figure 2.23). Code compliance tasks are assisted if we identify the WINDOW as a code complient facade. Work is underway to use these *usage attributes* when exporting ESP-r models. A future version of ESP-r will also make use of an opening type attribute (Figure 2.25) to help form air flow network (currently they are *only used for documentation*.

```
surface USAGE options

a FICTitious (not code complient)
b FRAME FACADE (code complient)
c FRAME OTHER (not code complient)
d GRILL SOURCE (conduit outlet)
e GRILL EXTRACT (conduit inlet)
f WINDOW  (facade code complient)
g WINDOW DISPLAY (code complient)
h WINDOW skylight (code complient)
i WINDOW OTHER (not code complnt)
j -   -  (somthing else)
```

Figure 2.23: Use for window.

Repeat this process for the window in Wall-5. Remember that you can rotate the wire-frame viewpoint to get a better view of your work.

For the door, use the `inserted into a surface->at base` option. Wall-2 is 4m wide and if the door is to be 1.0m wide (this is a medical facility) then the offset for the door will be 1.5m in Figure 2.26. When you have done this the display should look like Figure 2.27.

What might you look for in the interface at this point? The wire-frame image *complements* the text in the report below. The wire-frame and the report are designed to be used together.

For example, a surface in the wire-frame might look ok but the report indicates it is facing the opposite direction. The data included in the text feedback area report is similar to that included in a model QA report (discussed later).

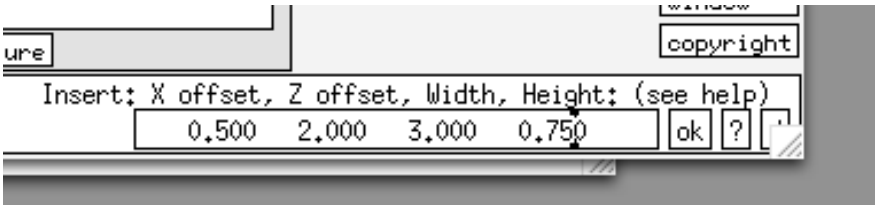


Figure 2.24: Size and offsets for window.

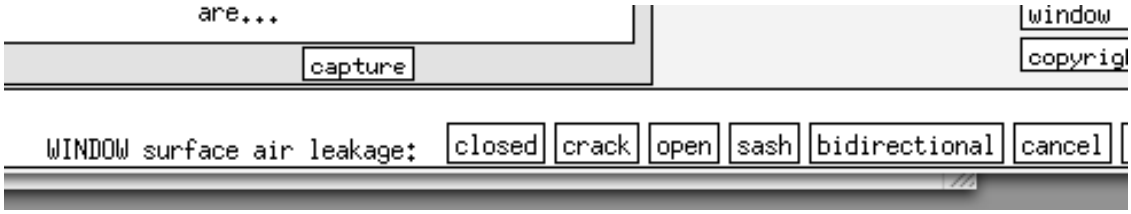


Figure 2.25: Window opening type.

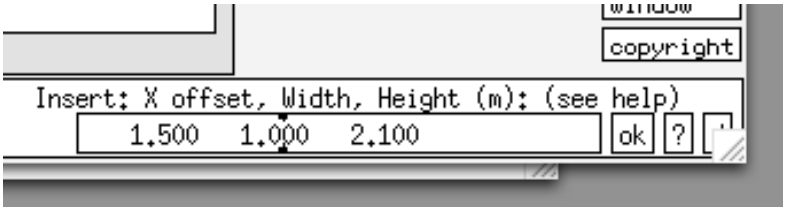


Figure 2.26: Door dimensions.

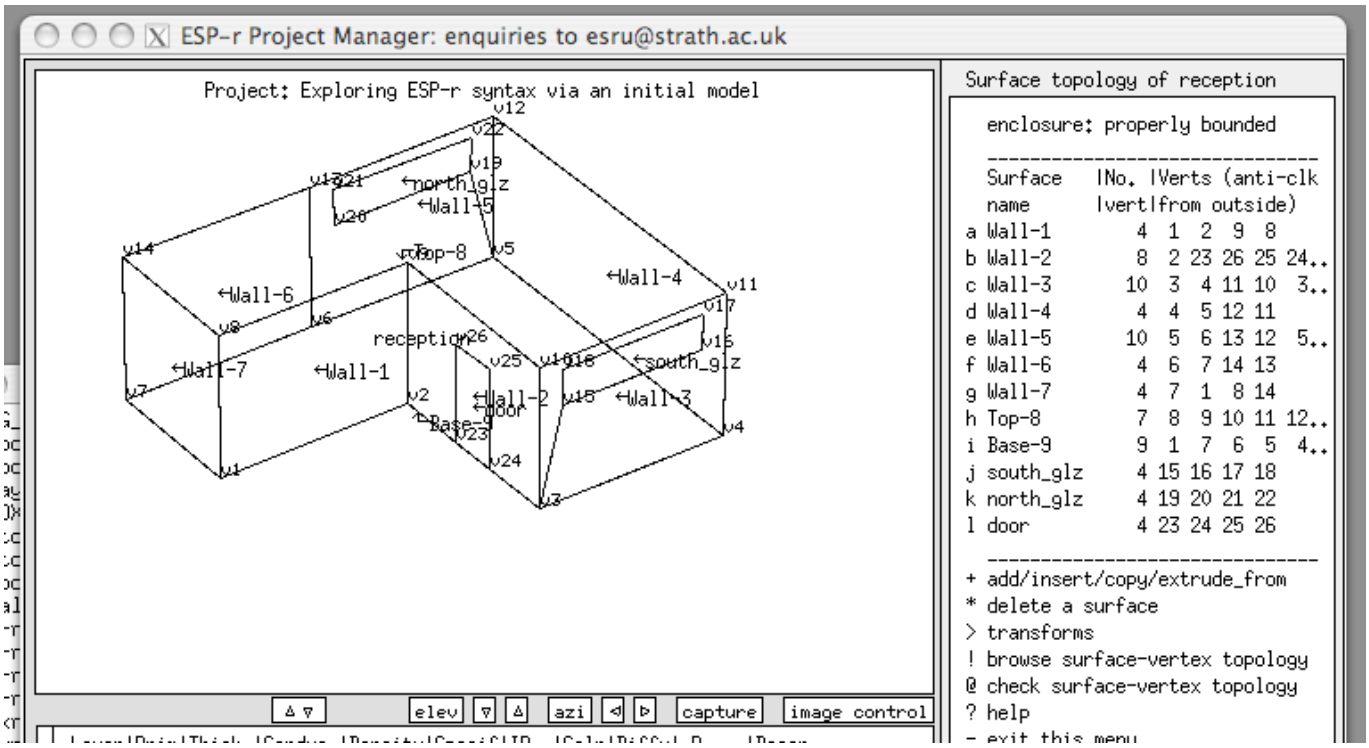


Figure 2.27: Reception after the addition of windows and doors.

Of course, at this point all of the initial entities have default names and many of their attributes are UNKNOWN. Surface attribution is what we will turn our attention to in the next section.

Completing attribution

Tactically, we would prefer that others find it easy to understand our models. If we can also reduce errors and speed QA tasks we have a winning combination. One successful pattern of model attribution is to begin with the names of the surfaces so that subsequent selection tasks and reports take less mental effort. Of course if you want to do it the hard way...

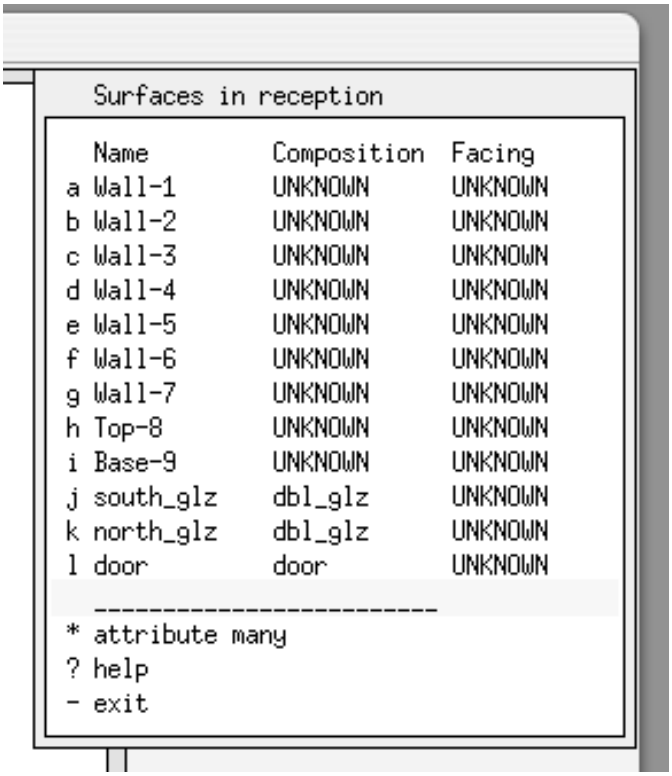


Figure 2.28: Surface(s) attribution.

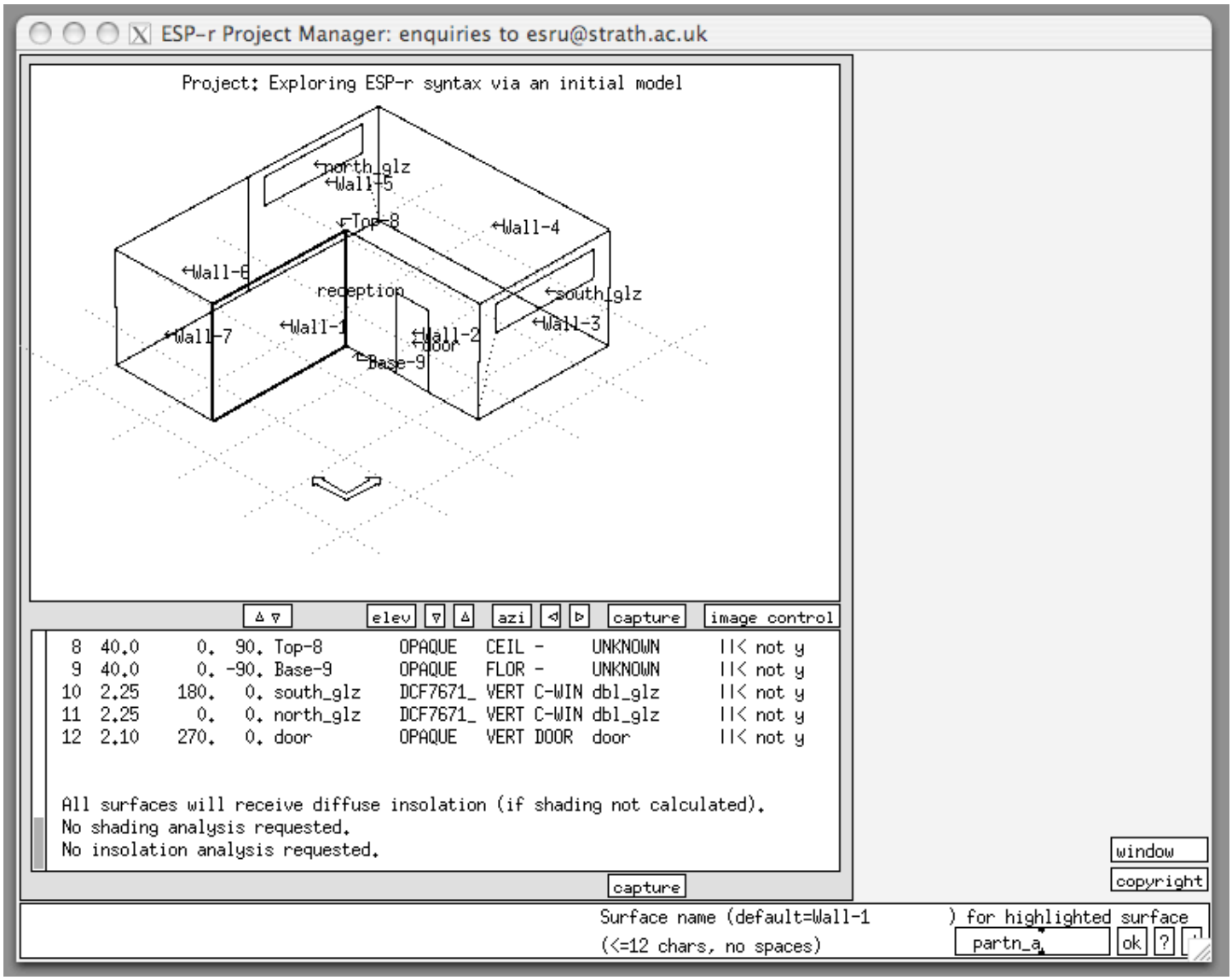
As shown in Figure 2.28, the surface attributes menu includes an `*attribute many` option (this uses a list selection dialogue described in the Version Appendix). You have a choice of attribution types `name, composition, boundary condition,` Select `name` and you will be asked to defined the name of each of the selected surfaces (note the surface is highlighted in the wire-frame drawing). After attributing the names the display will be updated as in Figure 2.29.

To name something is to own it.

Tactically we want to take possession of our models. Giving entities names is a key step *especially if it helps the design team*. The pattern used in this room is one of many possible patterns and the rule is roughly:

- names like floor, ceiling, entry_door have almost instant recognition
- if the floor or ceiling is represented by several surfaces then append a unique identifier e.g. floor_a, floor_b or floor_1, floor_2 (but the latter might be misinterpreted)
- walls that face the outside might indicate which way they face via north_wall, east_wall but if a model might be rotated such names can be confusing so front_wall is better
- partitions can be named based on the zones on each side e.g. kitchen_partn, coridor_partn, some prefer partn_a, partn_b.

The pattern above uses the name to give a clue about the location and composition of the surface. Note: ESP-r is looking for unique combinations of characters, so other languages are ok as long as the ASCII character set is used.



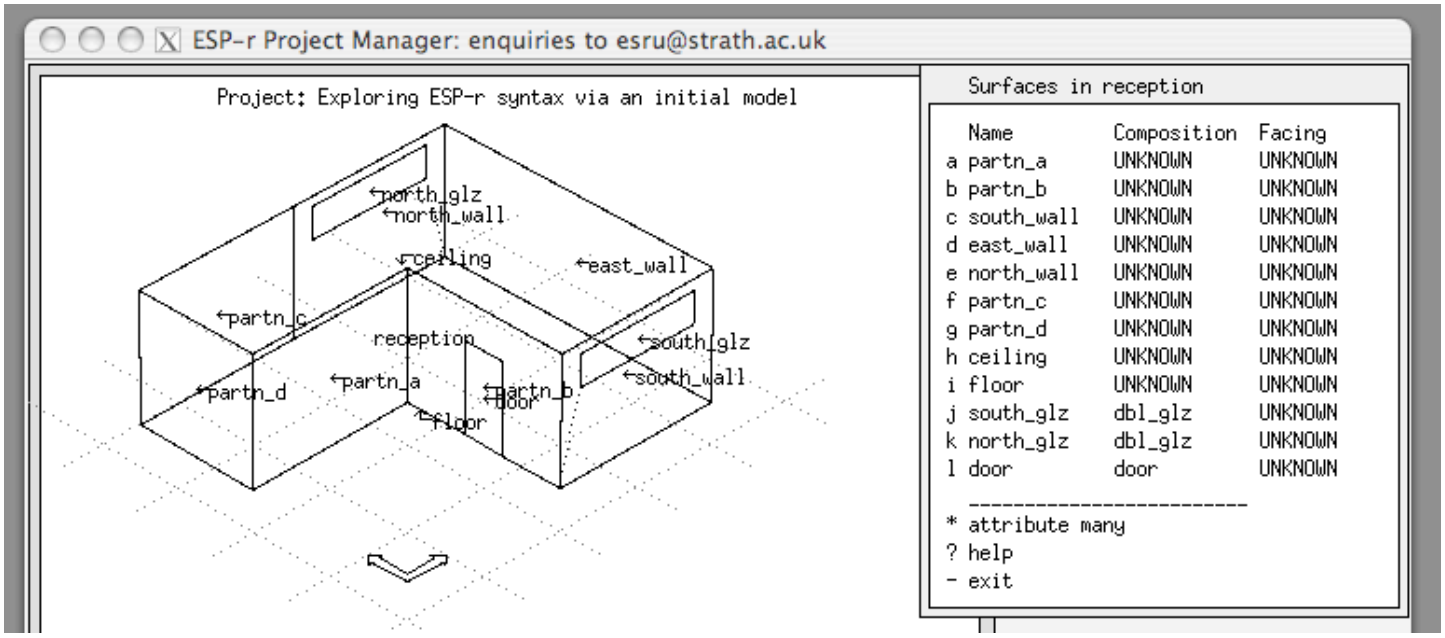


Figure 2.29: Surface naming.

Typically adding construction attributions, for those surfaces that you have an opinion about, would be the next task. Use the list of *useful* constructions you wrote down when you were reviewing the databases). Where several surfaces have the same construction use the `*attribute many` option. Otherwise select the surface you wish to attribute as in Figure 2.30 where all of the attributes of the surface are reported on and are available for editing. After you have selected the constructions the interface will look like the second part of the Figure.

Skip boundary conditions for the moment - there is an automatic process to assign this attribute later in the process.

The point of careful attribution is that the combination of the graphic image and surface attribution ensures that the model is correct. If something called `door` is composed of `concrete` and you see it as a horizontal in the image someone is likely to notice!

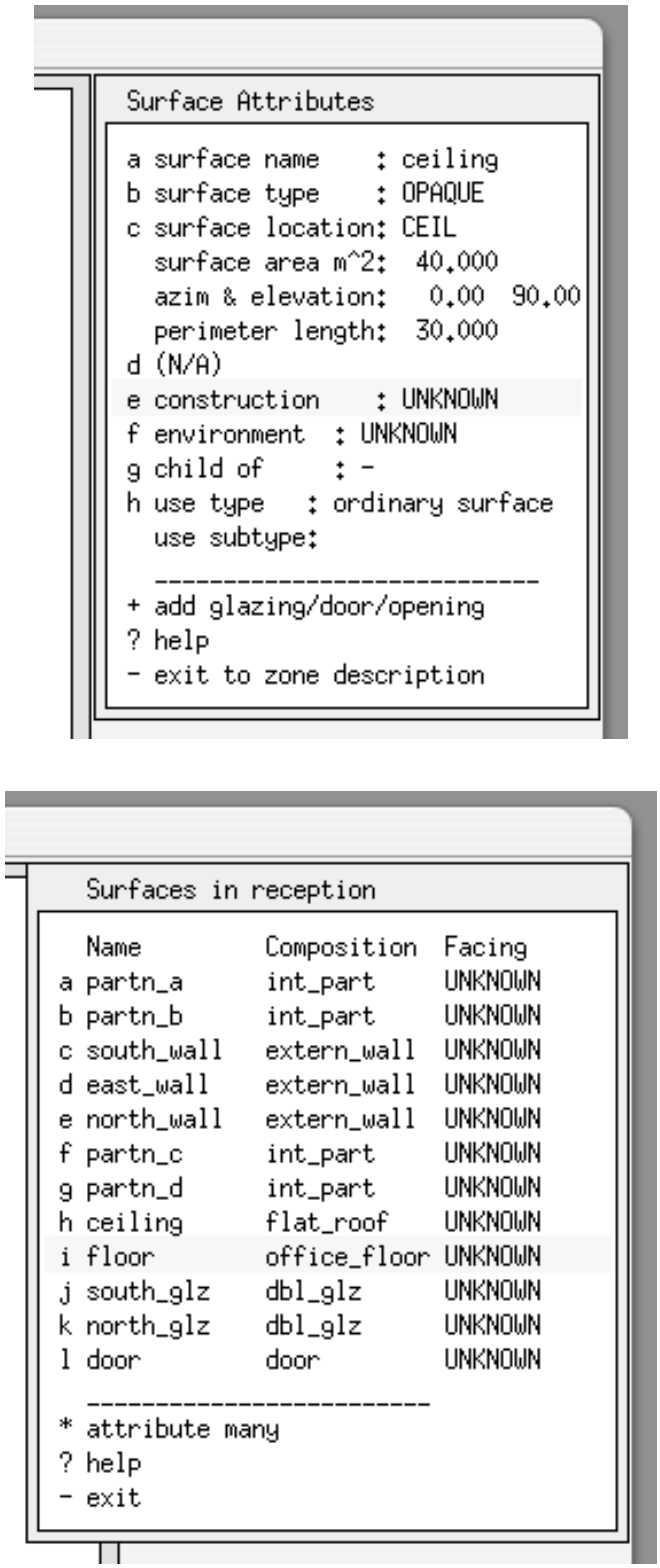


Figure 2.30: All the attributes of a surface and list.

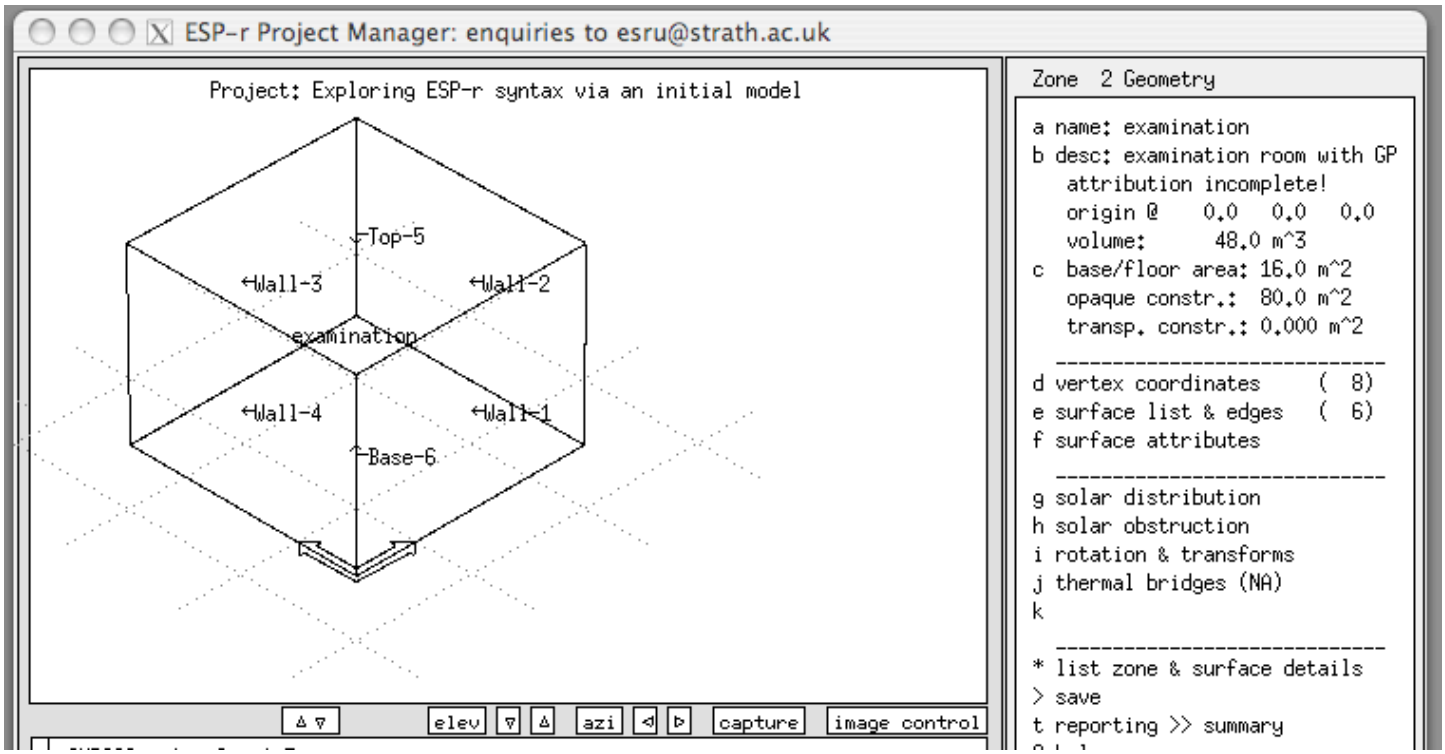


Figure 2.31: Initial box shape of examination room.

Adding the examination room

The examination room is rectangular in plan, but has a sloped roof and it shares partitions with the reception. There are a number of approaches to creating this zone:

- start with a simple shape and evolve it
- use the co-ordinates and link them together to form the surfaces of the zone

For purposes of this exercise we are going to start with a simple shape (a box) and transform it into the final shape. For additional practice look at Exercise 9 in the *Cookbook Exercises* volume. The transformation is going to involve changing two coordinates to elevate the roof, deleting a couple of surfaces and then copying surfaces from the reception. For purposes of this exercise it is an efficient approach and it will give you a chance to work with a range of transformation facilities.

Referring back to Figures 1.2 and 1.8 the initial box has an origin of 0.0, 0.0, 0.0 and a width of 4m (East along the X axis) and depth of 4m (North along the Y axis) and is 3m high.

So return to `Model Management -> browse/edit/simulate -> composition -> geometry & attributions.` Select the option to add a zone via `input dimensions.`

You will be asked for a name (say examination) and a description (say something like examination room with doctor and one patient during office hours). You will be offered a choice of initial form and this time choose `rectangular plan.`

You will be asked for the origin (0.0 0.0 0.0) and dimensions (4.0 4.0 3.0) and orientation (0.0). The result is

shown in Figure 2.31.

To make the roof sloped we need to alter the Z value of vertex 7 and 8 to 4.5m.

> If you cannot see the vertices in the X11 version select image control and toggle the vertices option. In the GTK version use the pull down view tab and toggle the vertices option.

Select the vertex coordinates menu option and pick vertex 7 and 8 and change the Z value to 4.5m. As you do this the wireframe image will be updated (see Figure 2.32).

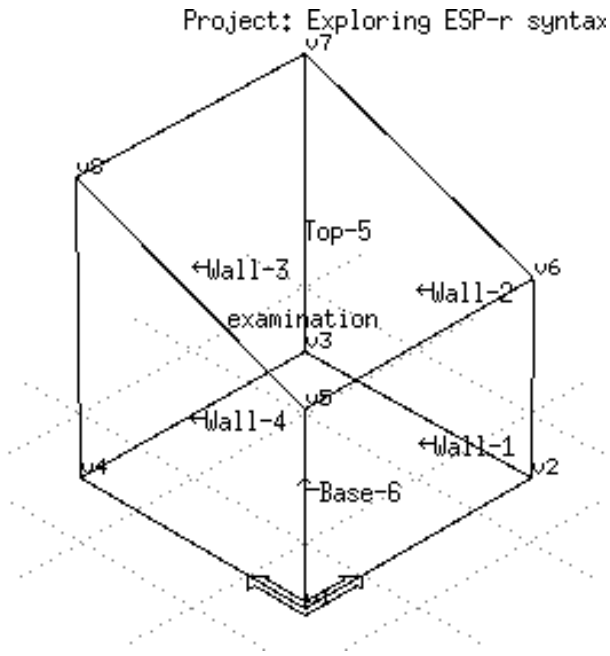


Figure 2.32: After editing vertices.

ESP-r has a rule: every surface has one boundary condition. So what is the boundary condition for Wall-2 and Wall-3? How does this compare to the initial sketch in Figure 1.2? The examination room has partitions adjacent to the reception zone as well as external walls and clear-story glazing.

One quick way to make this transform by first deleting both Wall-2 and Wall-3. The place to do this is the `surface list & edges -> add/ delete/ copy/ extrude_from` option. After you delete these surfaces the wireframe image will look like Figure 2.33.

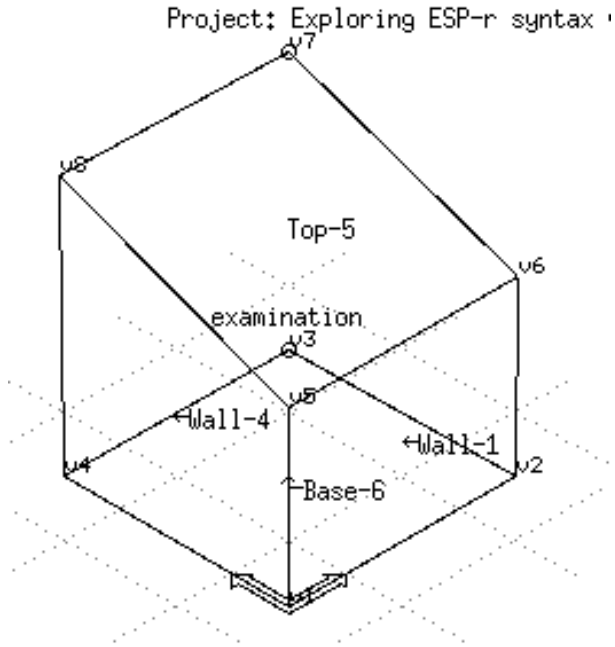


Figure 2.33: After deleting surfaces.

Our next task is to add surfaces to the zone by copying the relevant surfaces from the reception zone. Figure 2.34 shows the available options. We will be using several of these as we progress, first select `copy surface from another zone.` A list of known zones is presented (select reception) and then select surfaces part_a part_b and door.

You are then presented with a set of transform options for these copied surfaces. These options allow you to re-use existing surfaces in many ways without having to get out a calculator.

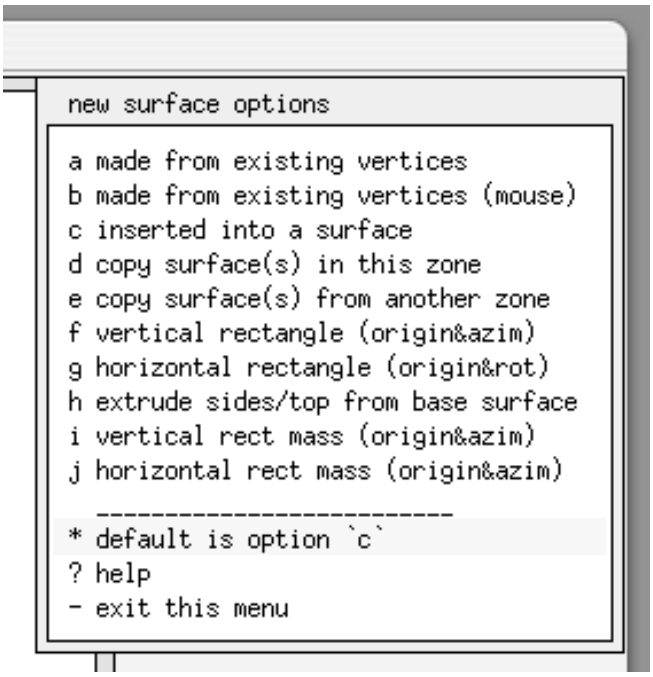


Figure 2.34: Copy options.

There is a rule in ESP-r:

> the order the edges of a surface are defined in tells ESP-r which is the outer face of the surface.

For example, part_a in reception had an azimuth of 180°. We need to reverse this and we select the `invert` option.

You will be asked to confirm this choice for the other two surfaces that you copied. You will also be asked if it is ok to update the edges of some of the existing surfaces to take account of the new vertices that have been included (say yes). After copying the surfaces your model should look like Figure 2.36.

Our next task is to fill in the upper external portion of examination. We already have most of the information we need. One of the surface addition options is from existing vertices. We must supply the vertices for the triangular shaped surface on the East side of examination.

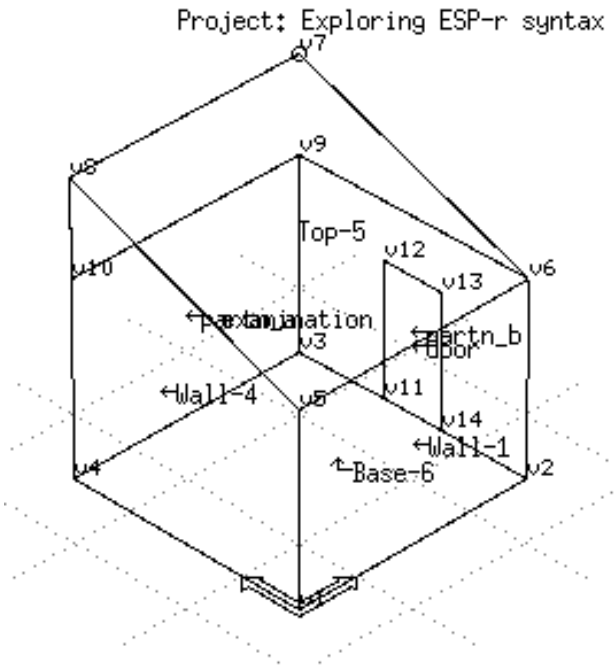


Figure 2.36: Examination room after partitions and door have been copied.

Another ESP-r rule is:

> If you see the outside face of a surface in the wireframe define the edges anticlockwise from the lower left corner. If you see the inside face of the surface in the wireframe then define the edges clockwise.

From this rule vertices 6 9 7 are of interest. Do the same for the wall on the upper North face. In the wireframe we see the inside face so the vertices are 9 10 8 7. This will become the frame around the clearstory window so name this surface something like north_frame.

Our final task is to insert the clearstory glazing. Select the `inserted into a surface -> as percentage` option and give 80%. This will centre the glazing in the surface and is a quick approach when the exact position of the glazing is not an issue. Attribute the glazing with dbl_glz.

Follow the steps you used in the first zone to complete the attribution of the composition of the surfaces that still have UNKNOWN attributes and to give names. Notice that there are fewer surfaces requiring attribution. Copied surfaces will be partially attributed. You should see something like 2.37 after you have completed the attribution.

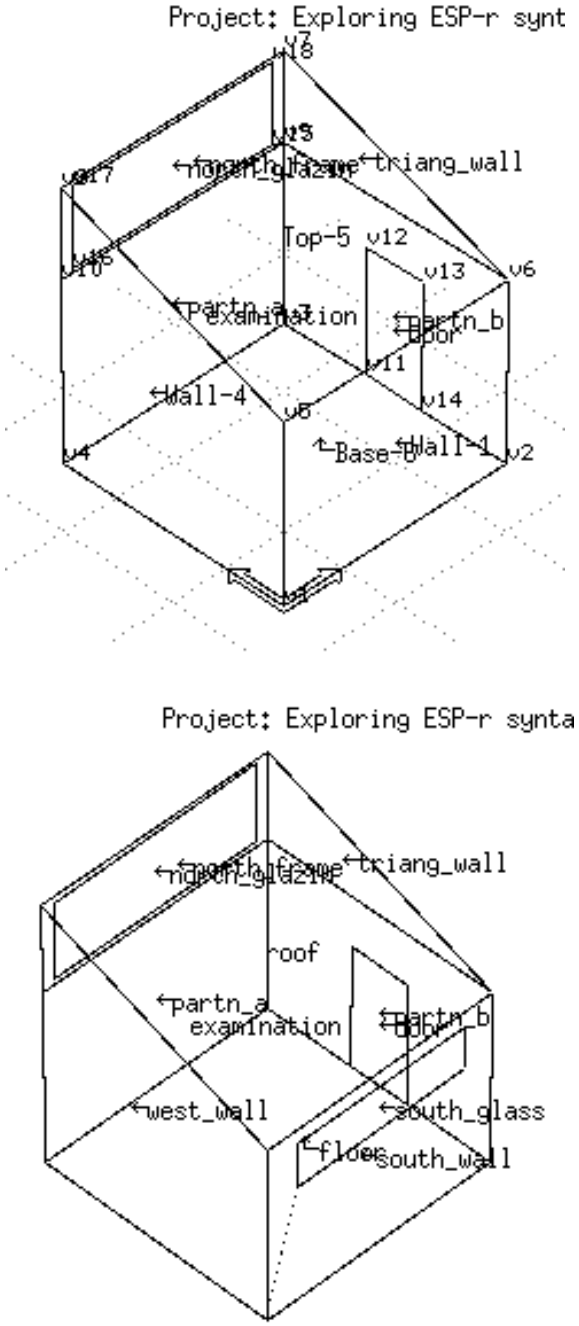


Figure 2.37: Examination room before and after attribution.

Now look back at your initial sketches. Is the model that you created correct? Look again...isn't there supposed to be a window in the south wall?

2.4 Model topology

Having completed the form of the zones there remains the task of defining the boundary conditions which apply to each surface in the model. There is an automated process for this and our next task is to use this facility. For additional practice complete Exercise 10 in parallel with reading this section. Go to `Zones composition -> surface connections & boundary` (see Figure 2.38). The option you will want to select (after reading a bit further) is `check via vertex contiguity` but first click on the `? help` option and read about the facilities.

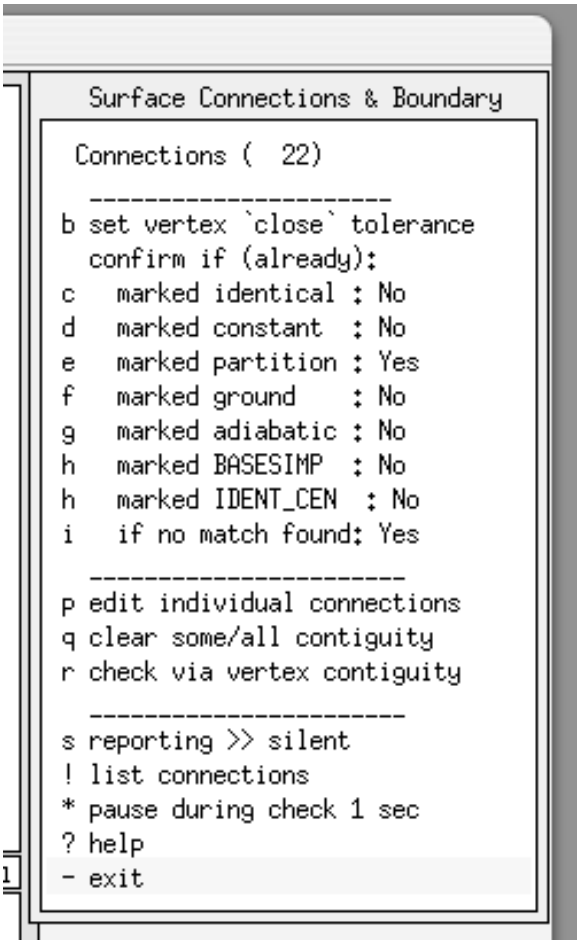


Figure 2.38: Topology options within the project manager.

The topology facility scans the polygons of a model looking for surfaces in various zones which are close matches in terms of shape and position and makes inferences from this to complete the boundary condition attribute of each surface. You control the tolerance and the extent of the search parameters and if the tool is unsure of what to do it will pause and ask for confirmation (Figure 2.39).

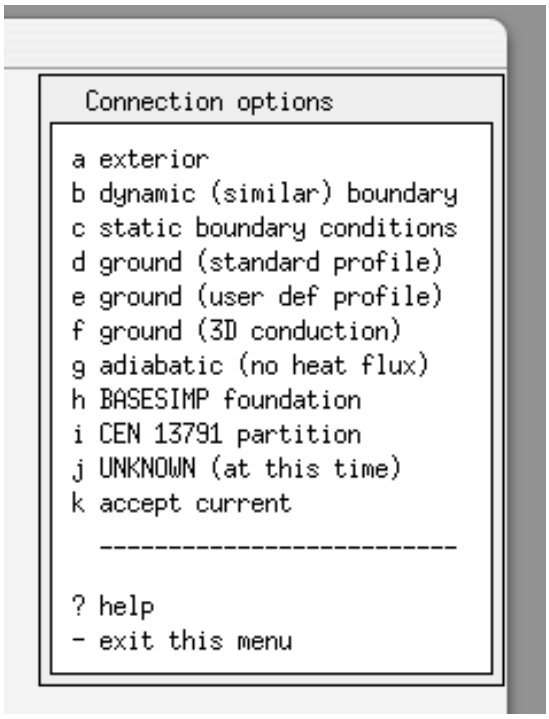


Figure 2.39: Topology checking confirmation options.

One by-product of checking model topology is that it checks every surface in your model in sequence so it makes a great way to review your model. If you periodically invoked the topology facility (say after adding two zones) you will have a chance to review the model at the same time it is searching for matches to the new surfaces you have added.

Up to this point the strategy has been to follow working procedures which help

us to create correct models. How do we know they are correct? One of the steps in checking the quality of our models is to generate a QA report and then review this against our initial sketches. Exercise 12 of the *Cookbook Exercises* volume is all about QA and this point in the model creation process is an appropriate time to complete that exercise.

In order to run a simulation, each zone in a model must include full thermo-physical data for each layer in each surface. To practice creating these zone files read and complete Exercise 14 of the *Cookbook Exercises* volume. If your initial zone geometry is properly attributed the task of creating these zone files will only take a moment for each zone.

3 Geometry alternative inputs

ESP-r offers several options for geometric input: creating rectangular bodies, extruding floor plans, working with polygons, clicking on points on a grid, clicking on points on a bitmap image (e.g. site plan, building plan or elevation) or importing CAD drawings.

It is up to you to select the approach or mix of approaches which are appropriate for your skills and for the model you wish to create. In planing your simulation tasks consider the regularity of the plan, the quality of the bitmap image and the level of clutter in the CAD file. A plan with a 1.3m x 1.7m repeating pattern will not easily fit within ESP-r's gridding options. A bitmap with only a few pixels per metre will be difficult to accurately select points on and a CAD model that includes thousands of extra surfaces for furniture might be a good candidate for converting into a bitmap. Also consider whether you might use the facilities discussed in this section to acquire critical points on curved elevations and in non-rectilinear plans.

This section focuses on clicking on points on a grid with the goal of creating the same model as the first exercise. Later there will be examples shown of using images from historical records and maps as the source of points in a model.

If you are using the Native windows graphic interface or the GTK version of ESP-r on any platform this alternative graphic input facility is not yet available.

The planning stage for re-creating the doctor office is essentially the same:

- review the available information
- establish the level of detail required
- sketch the model (use your previous sketch)
- identify critical dimensions and/or points on the sketch
- decide on the sequence of zone and surface creation

Begin with a review the model from the first exercise and find your notes from the previous exercise. The overall dimensions of the model are 8m (east-west) and 7m (north-south). Corners fall on a 1m grid. The exemption is the windows and doors but these will be added later. For this exercise use a 0.5m grid for generating the zones.

As with the first exercises, the identification of critical dimensions, perhaps by marking on an overlay of your initial sketch, *is also a key step for this alternative mode of input*. You will be

shifting your attention between the sketch and the screen so you will want to find ways to record your progress.

In this exercise you will extrude both zones in sequence and then use the normal geometric manipulation facilities to make the examination roof sloped and add the windows and doors into each zone. Typically one would want to create a sequence of zones (or even a whole model) in one session.

A skilled user might expect an average rate of one surface every few seconds and might input a couple of dozen zones in one session. This exercise is intended to help you acquire the skills needed to use the click-on-grid facility. It will likely take you several iterations to acquire the necessary habits.

Before using your keyboard and mouse have a read of the next pages and look at the figures. There are multiple steps involved. Those who plan their work and then implement the plan without interruption have the best chance of success.

3.1 To the keyboard...

Begin this new project by exiting any open versions of esp-r. Return to where you keep your models and then invoke esp-r.

To create a new model following the same process as you initially used:

- select `Model Management -> create new` with gp_grid as the root name.
- accept the folders for this new model fill in the high level description of the project
- fill in the site data

After the site details have been entered select `browse/edit/simulate -> composition-> geometry & attribution.`

Select dimensional input and provide the name and a description for the first zone (reception).

Lets start clicking...

To use the click on grid approach to geometry definition choose the `bit-map` option. This opens up an initial (blank) command window as shown in Figure 3.1 which allows you to select one of several pre-defined grids or to supply your own bitmap (scanned from a document or created from a CAD tool).

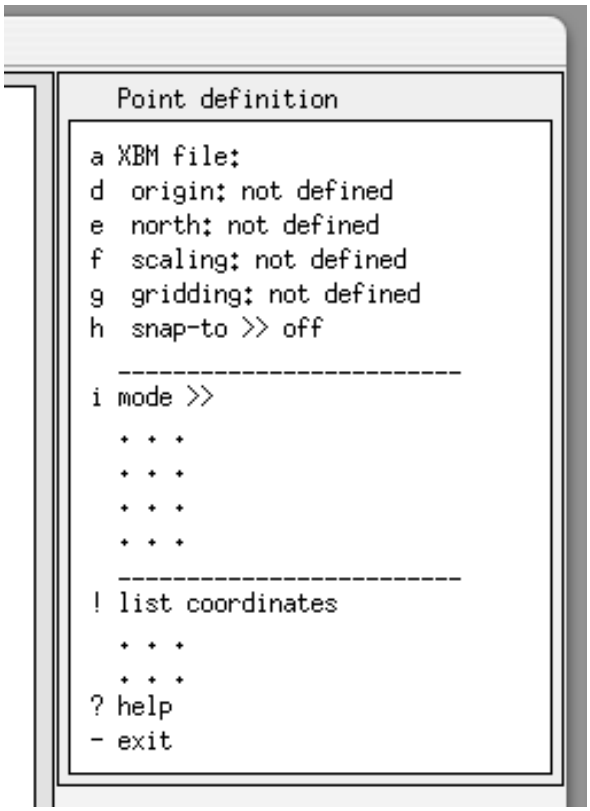


Figure 3.1: Opening menu of click-on-bitmap

Before selecting the grid file, adjust the size of the graphic feedback area to be closer to square. You might also use

the window manager to resize the Project Manager so that you have plenty of room to work. This will prevent you having to 'pan-around' the grid as you work.

For this exercise select the `large for gridding` option and accept the suggested file name in the dialog box. This option gives 23 horizontal and 17 vertical so considering each tick is 0.5m will give plenty of space.

The initial grid requires further information in order for you to use it as a basis for creating new zones:

- identify the origin (X=0.0, Y=0.0) typically slightly in from the lower left corner - say at the left + mark.
- define the scale of the grid by drawing a line of a know length. If each of the + are 1.0m apart draw a line from the left + to the fourth + and give the distance as 4.0.

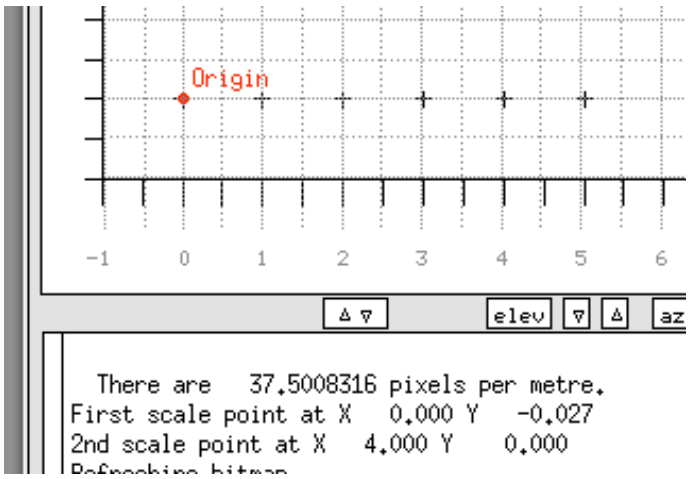


Figure 3.2: Origin and scale and grid

If you plan to define zones which extend into the negative X or Y dimension you would adjust the position of the origin to reflect this. Note that you can pan to the right and/or upwards as necessary but you cannot pan any farther left or down after you setup the initial origin.

Now that the scale is defined the 'real' grid can be overlaid by selecting the gridding option (choose 0.5m) and then turn on the snap-to option.

The mode >> menu option (Figure 3.3) lists a number of choices for entering data points. The first two options are useful for topographic/site data. The floor plan extrusion option is equivalent to the floor plan extrusion used in the initial exercise to create the reception zone. However, this time rather than typing in coordinates, you will be clicking on points on the grid you created.

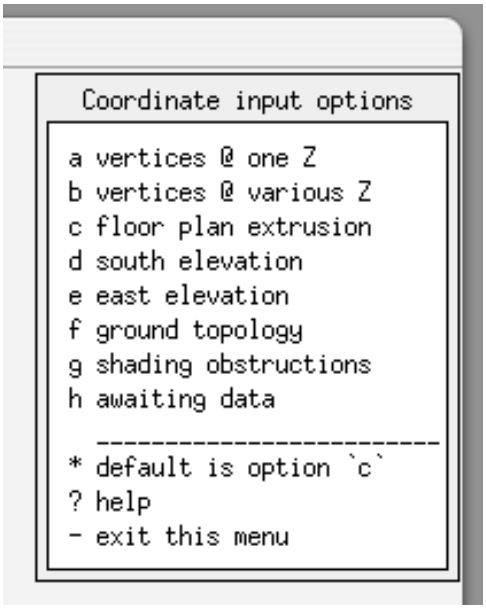


Figure 3.3: Which input mode

Once you have selected the floor plan extrusion input mode (see Figure 38) set the floor elevation to 0.0 and the ceiling elevation to 3.0 (to match Figure 1.2)

Just before starting to define points it is worth noting that you are able to interrupt the input process and move around a bitmap.

Use the `start option` to begin defining points in the same order you used in the previous chapter. Start at the grid nearest x0.0, y4.0 and then x4.0, y4.0 etc. until you reach x0.0, y7.0 (see Figure 3.4) after which you will type the character e to end the input.

With the snap-to option you only need to click near the point and it will snap to the nearest grid. If the snap-to option is off the Project Manager will accept the actual point where you click.

If you make a mistake or the point snaps to an incorrect grid point then immediately type the character d to delete the last entry (multiple deletes are possible.

After you have signaled the end of points for the initial zone (by typing the character e) you can save the zone data (if you did it correctly) or try again if you are not satisfied. While saving the zone data, an additional file is created to hold the 'topology' of the model and you can safely accept the file name offered for confirmation.

A new option `create another zone` will be displayed in the menu once the initial zone is saved. This can be used as many times as required to extrude zones (using the current floor and ceiling height attributes). In the current exercise the examination room needs to be added. It has the same initial floor and ceiling height as the reception zone so those attributes need not be altered. When you are ready, select the `create another zone option` and supply a name and description for the examination room.

Note that three of the corners of the examination room are at the same grid points as used by the reception and unless you specify otherwise, those coordinates will be used. Note also that the edges of the reception are still visible so that it is easy to create new zones adjacent to (including above or below) prior zones. Since you are extruding a floor plan you will proceed anti-clockwise from the origin typing the character **e** when you have done all four corners.

When you started clicking on each subsequent zone, a message is included in the text feedback to remind you of the keyboard control options. When you signal that you have finished selecting points for the examination room (by typing the character **e**) you will be presented with options to save or repeat the zone definition.

As the examination room is the last zone, exit from the click-on facility and you will be presented with a wireframe view of the zones you have created (Figure 3.6). These zones still require surface attribution as well as the addition of doors and windows and raising the roof in the examination room. Such tasks can be accomplished by using the geometry & attribution facilities introduced in the previous chapter. And this would also be a good time to revisit Exercise 12 in the *Cookbook Exercises* volume, generate a QA report and compare it with the original model.

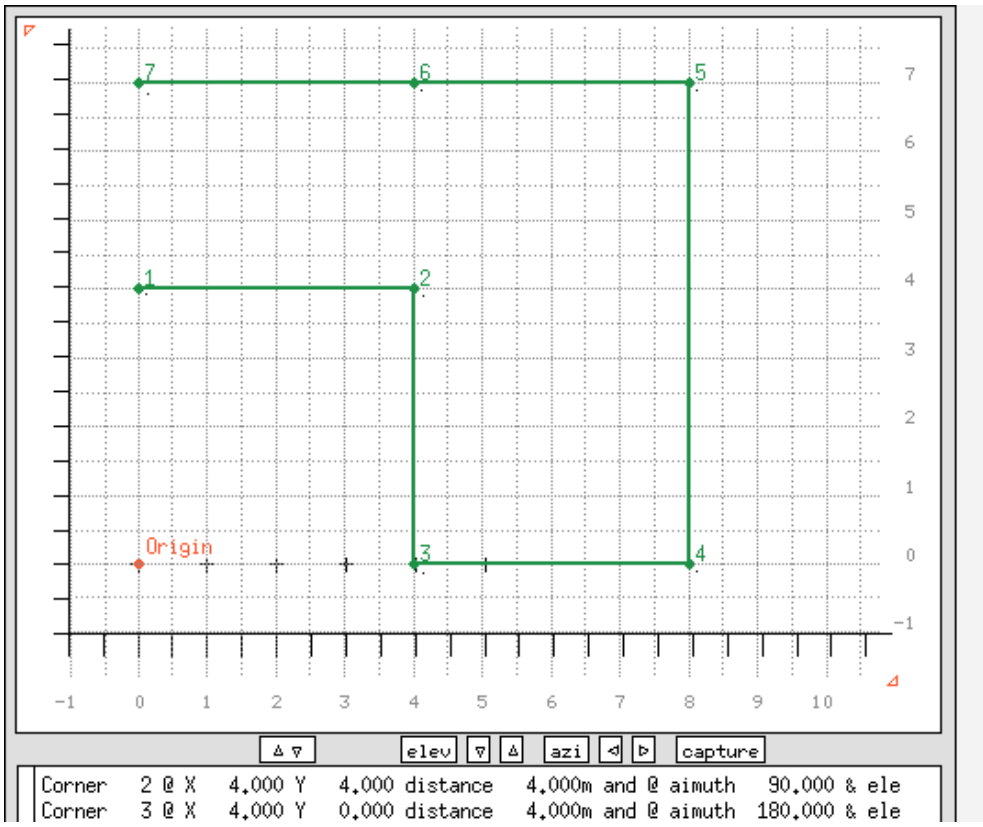


Figure 3.4: Just before finishing the first zone.

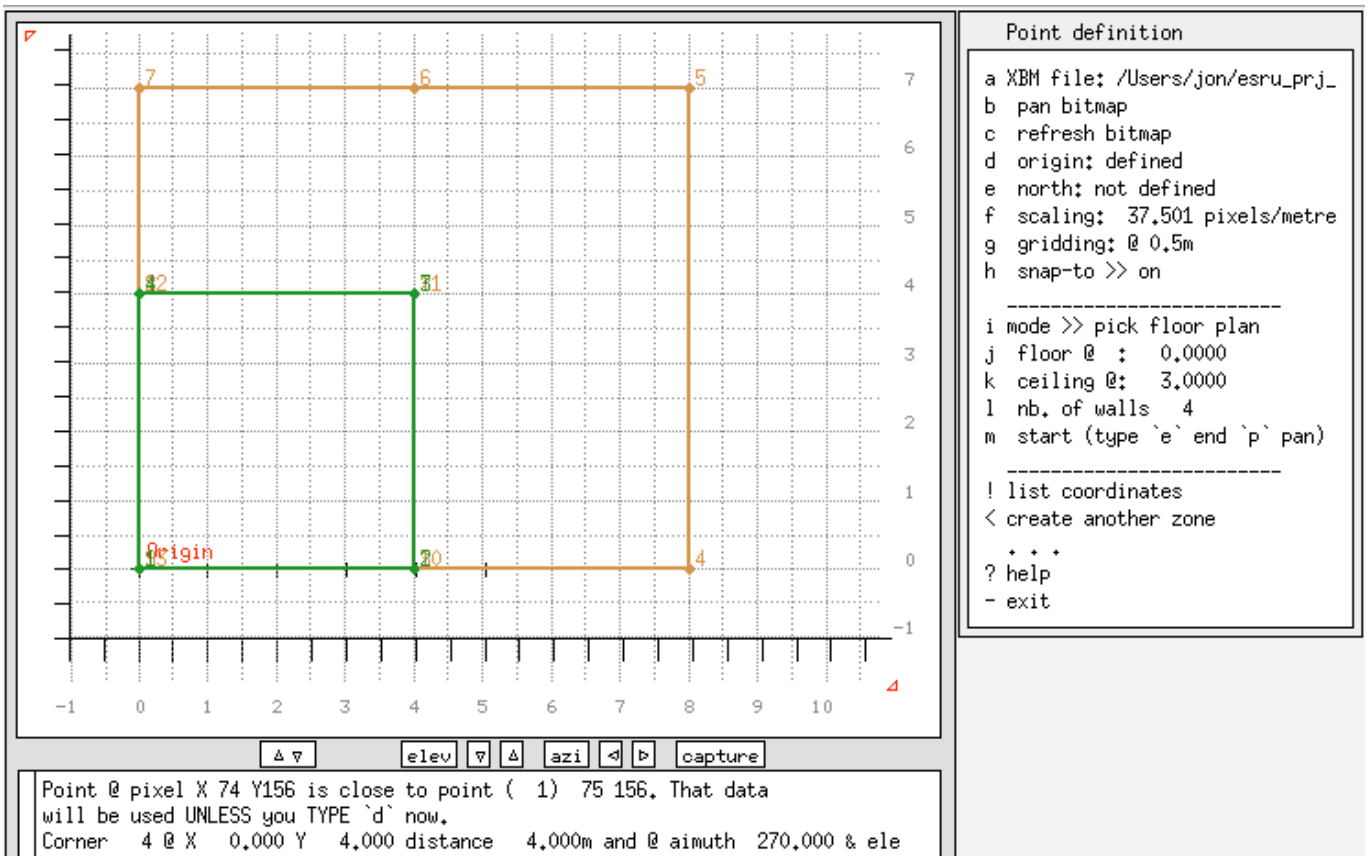


Figure 3.5: After finishing the second zone.

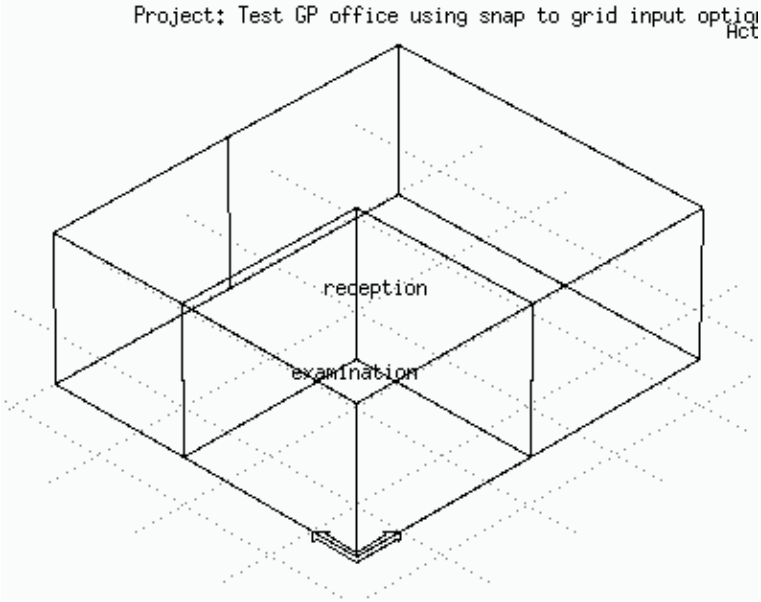


Figure 3.6: Another view of extruded model.

3.2 Clicking on a bitmap

A variant of the click-on-grid approach is to supply your own bitmap (plan/section/elevation/site-plan) and click on points found in that image to create one or more thermal zones.

In Figure 3.7 an UK Ordnance Survey map has been converted into an X11 bitmap file and used with the click-on-bitmap facility to create ground topography surfaces to associate with a model. Note that the solid lines (representing the contours selected) is an approximation of the contours on the map because ground representations supports several hundred surfaces rather than several thousand surfaces. The option used to within the bitmap facility were `points with different Z.`

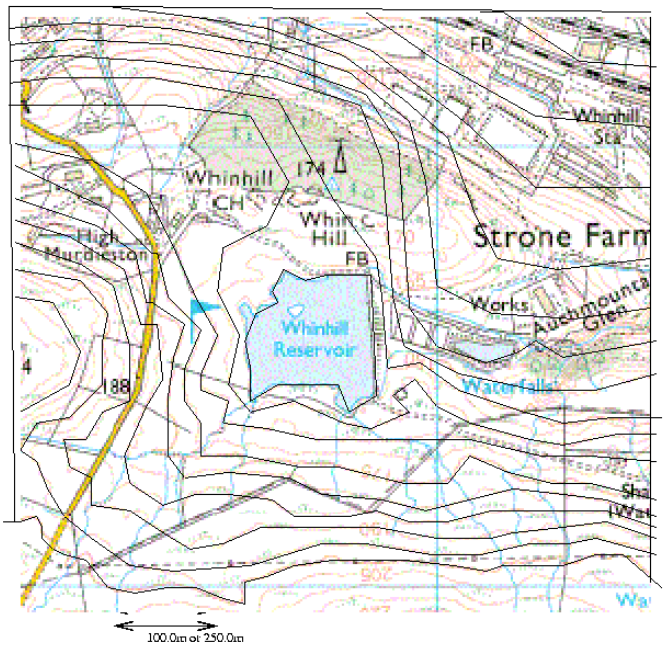


Figure 3.7: Using a contour map bitmap file.

Several tactical warnings should be given:

- try a small portion of a map until you are comfortable with converting such contours into surfaces.
- there is an automated triangulation facility but many users report that complex arrays of points can be

problematic.

- using a click on bitmap approach is no excuse for skimping on the planning of your model!

It is far to easy (bitter experience) to get carried away with the clicking and

- include more complexity than necessary
- get lost while doing the clicking
- collect points in such a way that the time taken to post-process the surfaces is greater than that associated with the clicking.

So ALWAYS...

- mark-up your image with indications of the critical points you want to capture and the bounds of the zones you will be creating,
- if you have bitmaps for more than one level make sure you can keep the points in-register and that the scale can be setup to be equivalent,
- sketch your model in three dimensions so that you can plan how partitions between zones work,

If you have ceilings and/or floors that are not level there are several possible approaches, some of which will save you time and some will not. Experiment with constrained models to see what approach to the click-on-bitmap works best for you. Keep a note of what works so that you can follow this tactic when confronted with a similar project in the future.

3.3 Examples of approaches to take

Once you have the requisite skills it should be possible to create substantial models with some speed. In the examples below there were two issues - creating models which the client would recognise and also models which captured the spacial characteristics of the building.

An example of using a scanned image to import information that is only available in hard-copy form (no CAD data). The theatre shown in was initially constructed in the mid seventeen hundreds and the last set of drawings available were from the mid nineteen seventies. It is also notable in that almost nothing is rectilinear.

The mode in Figure 3.9 was initially composed by clicking on a set of points from Figure 3.8 which were placed in dummy zones. Surfaces and points from those dummy zones were then used to build up the model zones. There was a detailed plan of the zoning worked out prior to the use of the click-on-bitmap facility. Even with this, considerable care was required and post-processing and reconciliation of the partition surfaces was required. This is an example of what can be accomplished by an experienced user and is just the sort of project which would be cruel and unusual punishment for a novice.

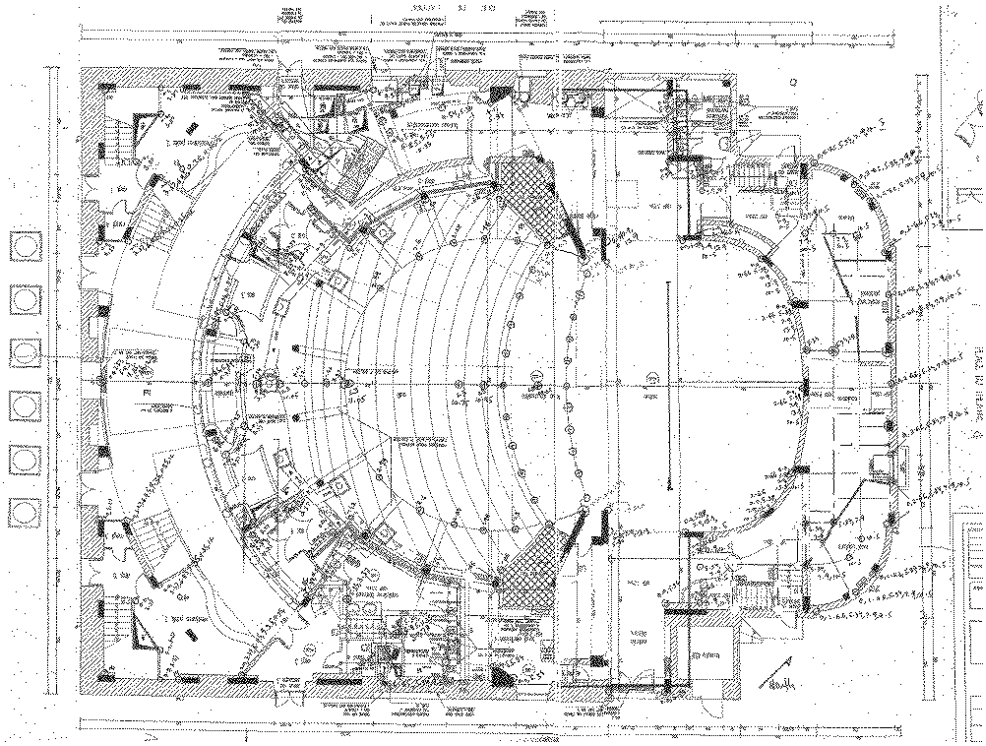


Figure 3.8: Bitmap image from historic building.

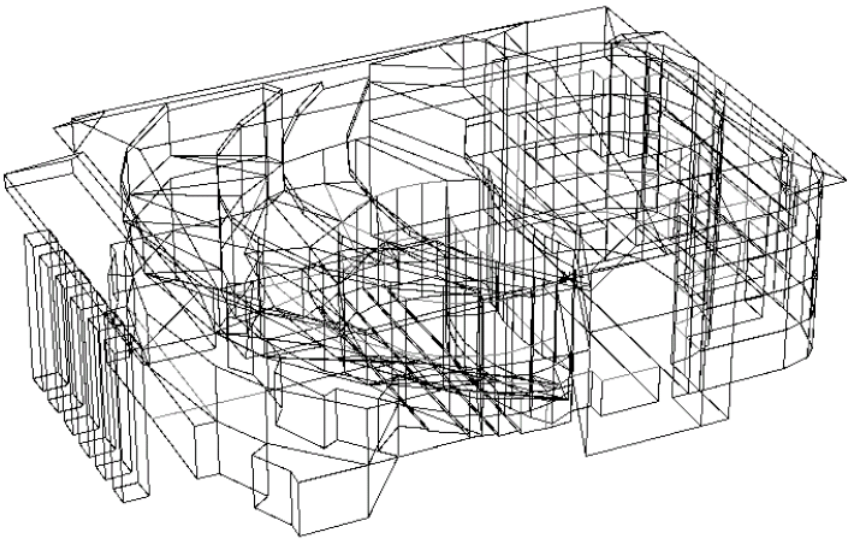


Figure 3.9: Bitmap image from historic building.

The model in Figure 3.10 and 3.11 was largely composed by a clicking on a mix of Ordnance survey maps, old planning documents and drawings. The model, including zone geometry, shading obstructions and ground topography was created and initial simulations commissioned within 12 hours of the completion of the simulation planning phase. It would have been faster, but one drawing was off-scale and several zones had to be adjusted to bring them into alignment.

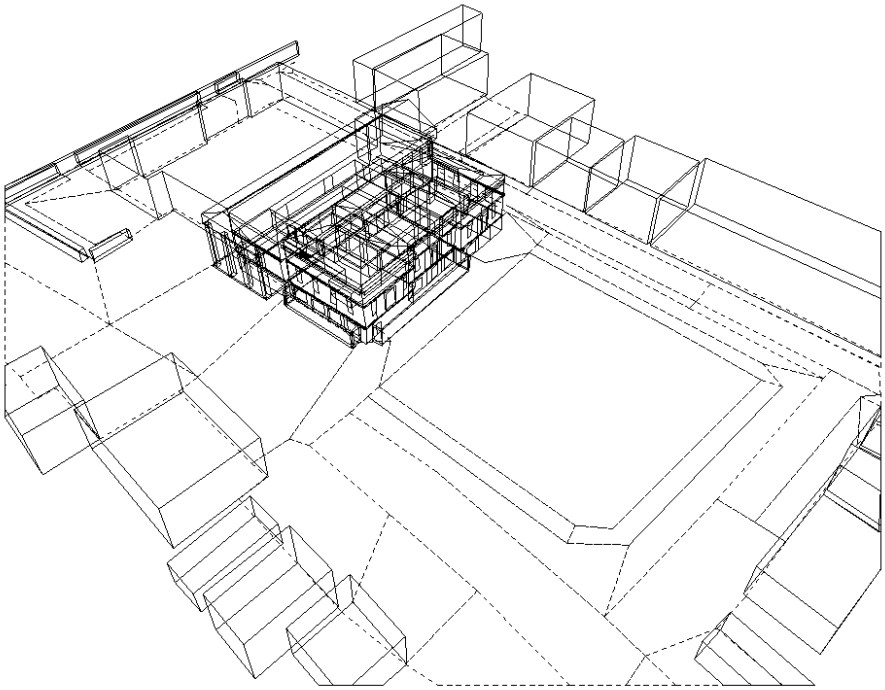


Figure 3.10: Layout of a museum and park.

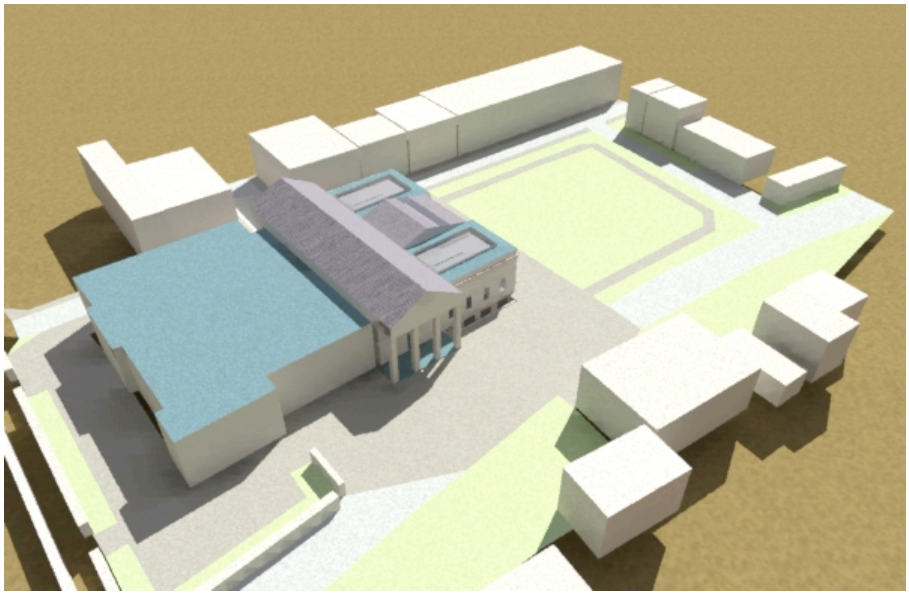
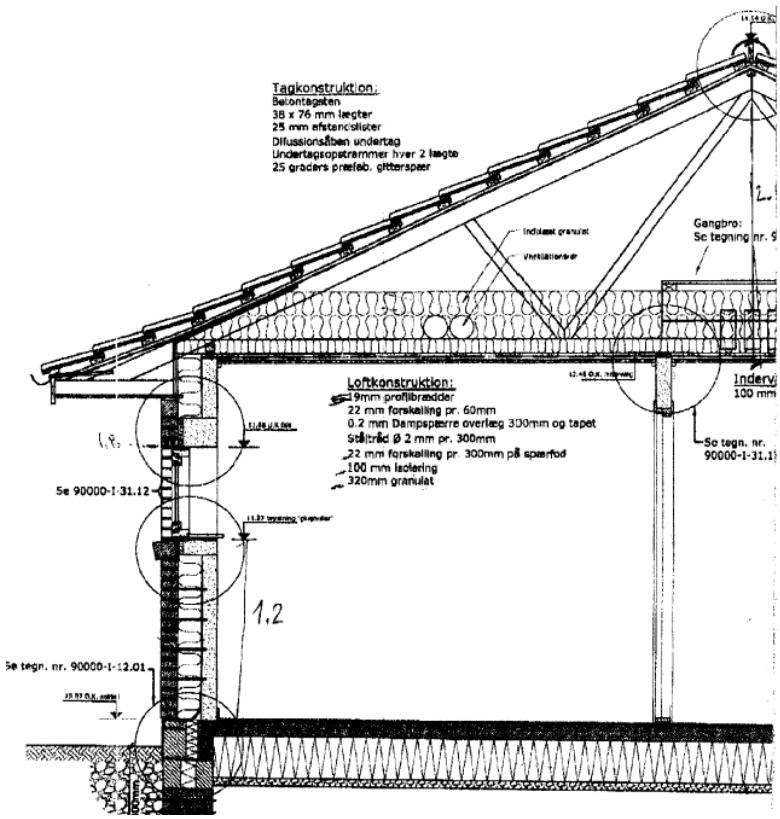


Figure 3.11: Rendering of a museum and park.

If you want additional practice (highly recommended) try to complete Exercise 11 in the *Cookbook Exercises* volume. It describes a four zone model to create. See how much time it takes you to create that model. There is no exercise that explores clicking on a plan image from a CAD tool - make up your own exercise!

These facilities are useful for many models and the fact that they have not been ported to the GTK interface is a limitation which needs to be addressed. The next chapter approaches the topic of model geometry from a different perspective.

4 3D modelling



Project: Abstract Version LaegehusStenloseSyd

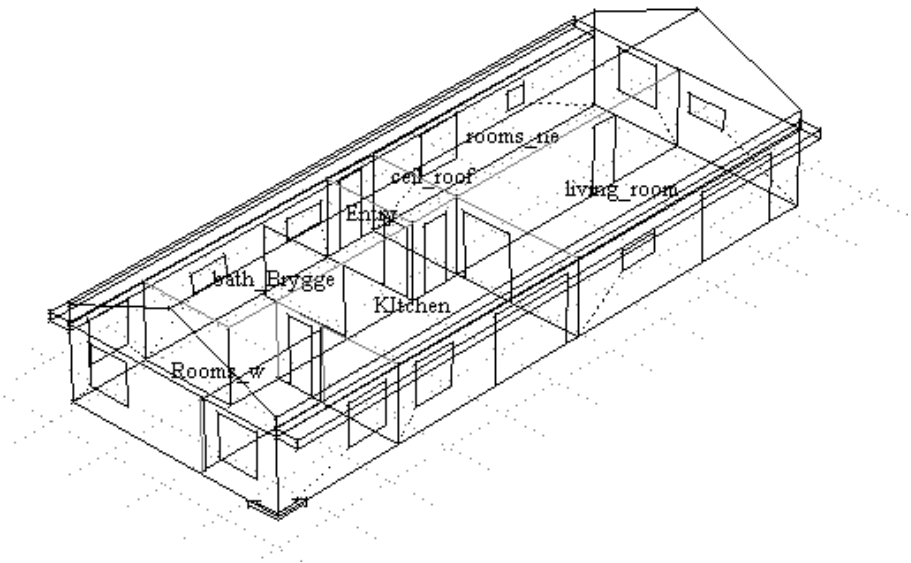


Figure 4.1 Section and view of house

The geometric forms discussed thus far use polygons as the building blocks of our virtual built environment. While modern (thin) constructions, such as those used in Figure 1.5 are often approximated by conventional polygons, historical buildings (see Figure 1.6) and the thick walls and insolation seen in Figure 4.1 highlight the limitations of this convention.

Focusing on Figure 4.1, there are several aspects of the section which need to be considered during the planning and creation of models:

- the thickness of insulation is substantial and at the edge is somewhat less thick
- the wall section comprises a number of material types
- there is an overhang which will act to shade the facade of the building
- the overhang is thermally isolated from the air within the roof space
- the overhang forms a boundary for the upper section of the wall.
- there appears to be an air space below the tiles which is separate from the air within the roof space.
- the portion of the roof with the tapering insulation is in direct contact with the layer of wood below the tiles
- it is not clear from the section whether the roof space is well ventilated.
- the area at the top of the insulation layer is somewhat different from the surface area at the ceiling.

The list of bullet points could be much longer. We need strategies for ranking thermophysical issues and deciding **what** needs to be included in our model(s).

4.1 Modelling approaches

The geometric and compositional resolution of a simulation model depends on the questions being asked and the resources available. Some thermophysical relationships may require simplification and other might not be possible to represent within our model. All virtual environments are abstractions. Simulation tools support one or more levels of abstraction for each of the domains they solve. Options allow experts freedom at the cost of a steep learning curve for novices.

A *tactical approach* to simulation uses the planning phase to constrain options. The following is *one possible ranking* of what to preserve while abstracting a design into a model:

- the volume of air
- the slope of the roof
- the location of mass within the roof
- the surface area in contact with the air

A low resolution model might treat the overhang as a solar obstruction and ignore the different thickness of the insulation. It might assume the air is well mixed within the roof space (i.e. there is no temperature stratification). It would also not explicitly represent the overhang as a boundary condition for the upper portion of the wall.

A medium resolution model might subdivide the surfaces to represent full thickness and the partial thickness insulation and extend the roof zone to

allow it to form a boundary at the upper wall section.

A high resolution model might represent the overhang portion of the roof as a separate zone or zones because there will be times when the air temperature in the overhang is different from that of the main section. A high resolution model would have an upper and lower section so that the temperature near the peak of the roof can be different from the air adjacent to the insulation layer.

By default, ESP-r assumes one-dimensional conduction. The thickness of the insulation in Figure 4.1 poses a challenge in comparison to a suspended ceiling. Do we choose to ignore the ceiling thickness when defining geometry? Use of physical co-ordinates would help to preserve the volume of the roof space and the surface areas and takes more time to describe.

Many example models distributed with ESP-r *appear* to ignore the thickness of partitions while other example models indicate a separation between rooms. Such differences are typically related to the method of data input. Geometry digitized from CAD drawings will have rooms separated in space. Zone geometry created from dimensioned data or sketches may tend to have partitions at the centre line of walls and the inner face of a faade.

The good news is that such differences typically have little or no impact on predicted performance. The solver knows from the thermophysical composition of the partition whether or not the two faces are separated in the coordinate system or lie in the same plane. There are exceptions which are covered elsewhere.

If the user is constrained in time, the user could form the base of the roof space by copy and invert the existing ceiling surfaces and then create the sloped roof above that (see Figure 4.2). This approach results in a model that is crude visually. The surface of the roof is at the correct slope, but the height of the building is not correct.

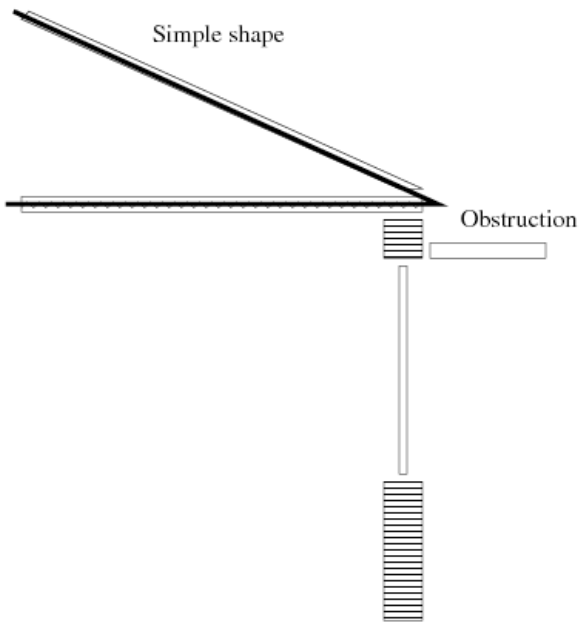


Figure 4.2: Constrained model

Given a bit more time the user could add a number of perimeter surfaces so as to raise the roof (at the expense of an inaccurate air volume) as in Figure 4.3. Such trade-offs might not alter the overall performance of the building but may be useful to make the model less crude visually.

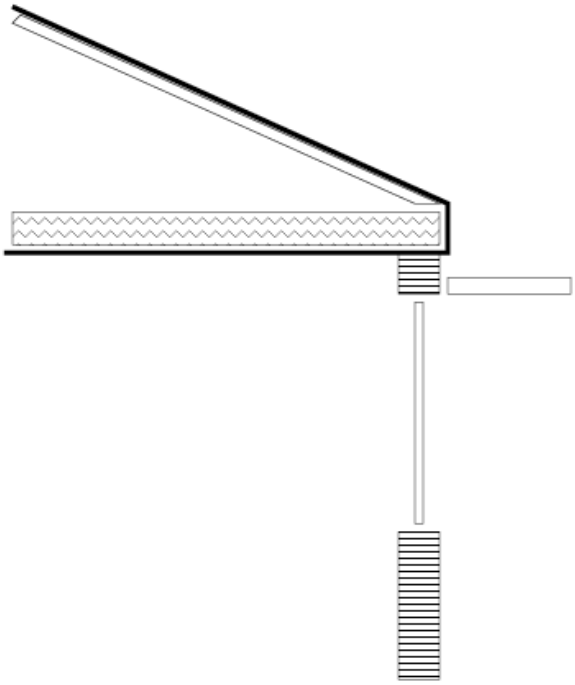


Figure 4.3: Variant placing roof at correct height

To move from treating the ceiling as a geometrically thin plane towards a model that uses explicit coordinates requires an additional step. The initial copy and invert of surfaces is followed by a surface transforms (along the surface normal). This facility, and several other types of rotation and transform are available in the interface (see Figure 4.4). If you find it difficult to see all of the lines or labels in the wireframe view use the GTK view facility to rotate or highlight portions of the model (see Figure 4.5).

If there is a gap between the roof zone and the occupied zone (as in Figure 4.6) this might be visually confusing to some users and their clients (if the ESP-r model was exported to Radiance). This gap could be filled with an appropriately sized solar obstruction block.

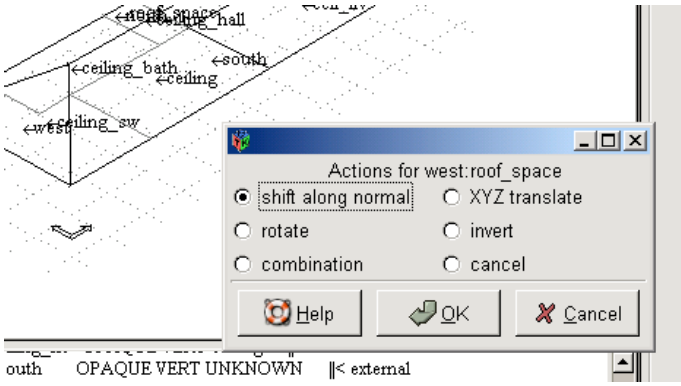


Figure 4.4: Surface transform options

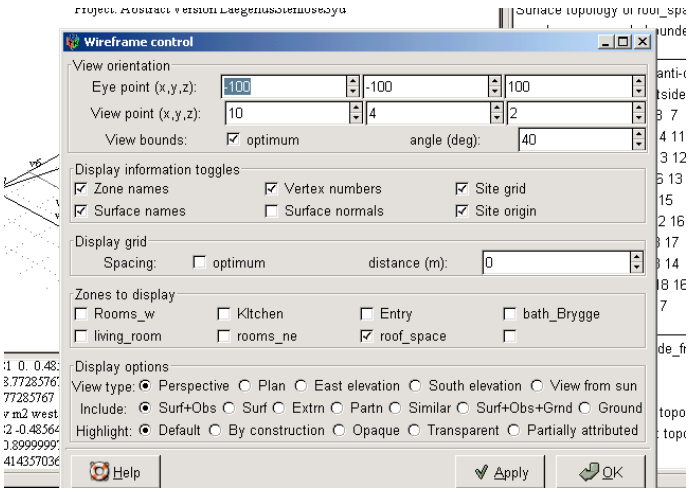


Figure 4.5: Wire frame control dialogue.

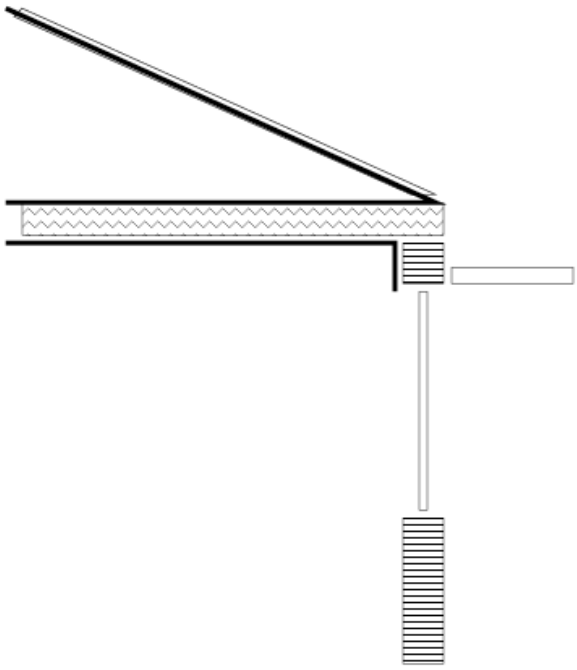


Figure 4.6: Variant using coordinates.

The above approach might be a reasonable thermophysical representation. It is, none-the- less quite abstract for users expecting a CAD representation. To approximate the 3D geometry of an actual roof requires that the initial copy, invert, transform of the ceiling polygons is followed by the addition of surfaces to represent the roof overhang as in Figure 4.7.

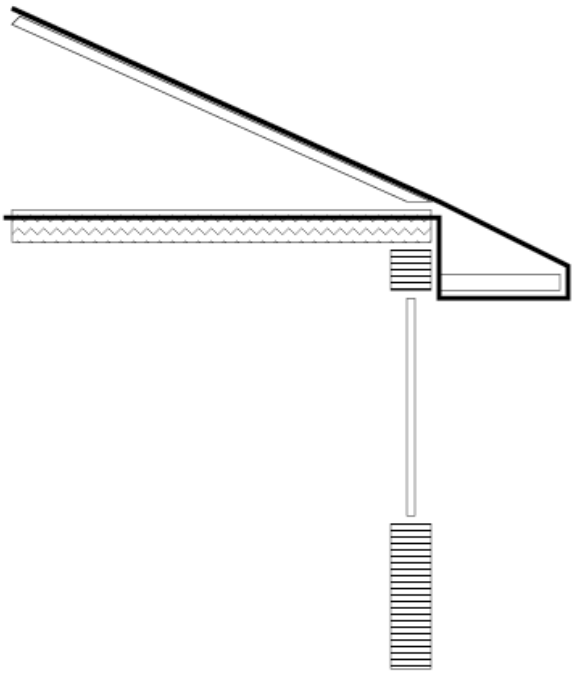


Figure 4.7: Variant extending overhang.

Note that the surfaces forming the overhang do not (in the current version) shade the wall. Shading requires the use of shading obstruction blocks (as included in the earlier figures).

And this more-explicit approach introduces a problem for the occupied space. The overhang, as drawn in the building section is in contact with the upper portion of the wall. The geometry of the walls should be adapted to sub-divide the wall into surfaces that face the outside and surfaces which connect to the overhang. Clearly this would be tedious to retrofit into the existing zones shown in Figure 4.1.

As mentioned in the introduction, the air within the overhang could be at a different temperature than the roof space. If such temperature differences were an issue, the overhang could be represented as a separate zone as in Figure 4.8. This overhang zone could wrap around the main roof space zone (one overhang zone could represent the overhangs on the North, South, East and West. Again the existing wall will have to be revised to represent the connection to the outside and the connection to the overhang.

One could be pedantic and argue that the temperature of the North overhang might differ from that of the South overhang and require separate zones. Few users would be confronted by projects where such detail is warranted.

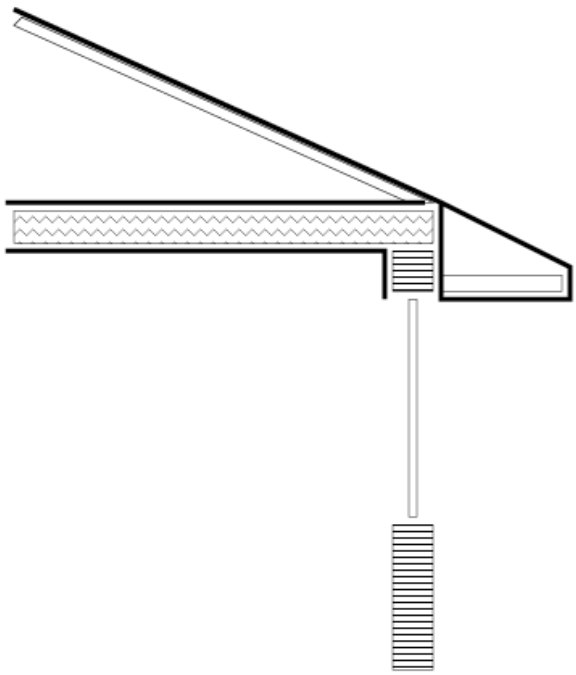


Figure 4.8: Variant with separate overhang zone.

Investing resources to increase model resolution is a decision that should not be made lightly. Some differences in performance predictions can be subtle rather than dramatic. A user who wishes to explore this could define variant models at different levels of resolution to test the sensitivity of predictions.

4.2 Steps to create a roof space

A *tactical approach* to simulation reuses existing entities where possible. This roof is one example of using the existing ceilings to compose the base of the roof space. Experienced users plan their work for maximum re-use!

A low resolution model of a roof space over the occupied rooms shown in Figure 4.1, and making use of the existing ceiling surfaces, and which follows the pattern of Figure 4.2 has the following critical dimensions:

- the height at the peak of the roof 4.35m
- the lower face of the ceiling 2.35m
- the width of the overhang 0.6m

The following sequence will minimize keystrokes and limit the risk of error. Other sequences are possible - so try variants until you evolve a sequence that works for you.

Enter the menu `browse/edit/simulate -> composition -> geometry & attribution` menu and select each of the existing zones and review their contents and ensure that the ceiling surfaces are attributed (to save time in later steps).

Next in the `geometry & attribution` menu select `add/delete/copy.` After electing to `add` a zone, select `input dimensions` and enter a name such as `roof_space` as well as a description to clarify to others the intent of this zone.

Because most of the initial surfaces will be borrowed from the existing zones it saves time to use the `general 3D` option for the initial shape.

The initial X Y Z position of the first surface is not applicable so just accept whatever is included in the dialogue because this surface will be deleted later on. Ignore the warning about the volume of the zone being zero.

An initial wire-frame image with a single surface will be displayed (see Figure 4.10). Your first task is to go into the `surface list & edges` menu and select `+ add/insert/copy` surface. Use the `copy surface(s) from another zone` option several times to form the base of the roof_space zone. It does not matter what order you copy the surfaces as long as your work pattern avoids duplication and ensures that all of the ceilings are copied.

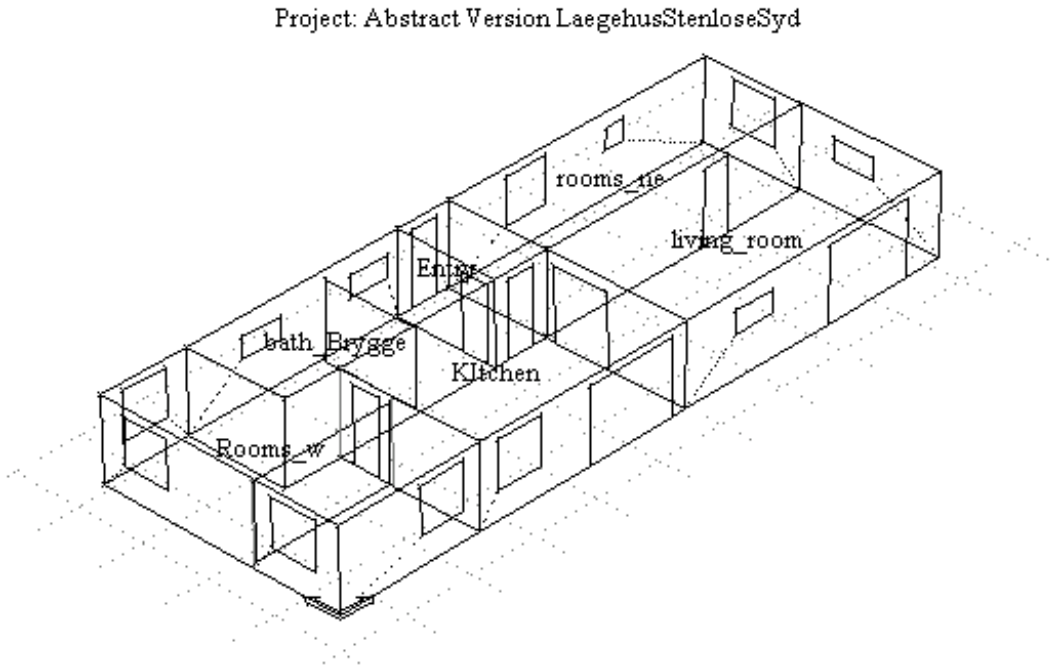


Figure 4.9: Occupied rooms in house (to place roof over).

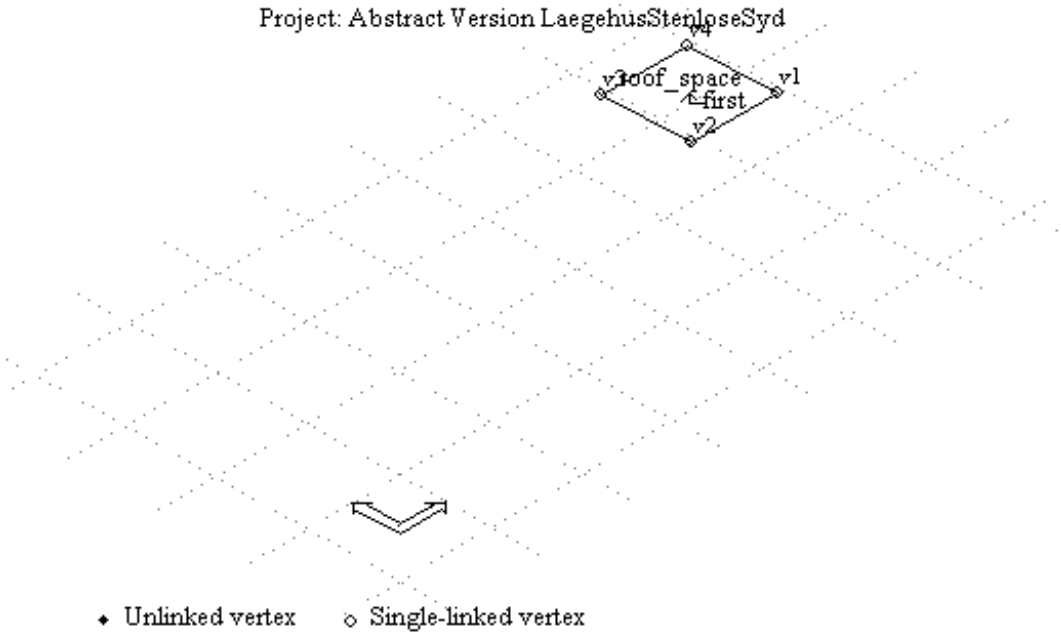


Figure 4.10: Initial dummy surface in roof_space.

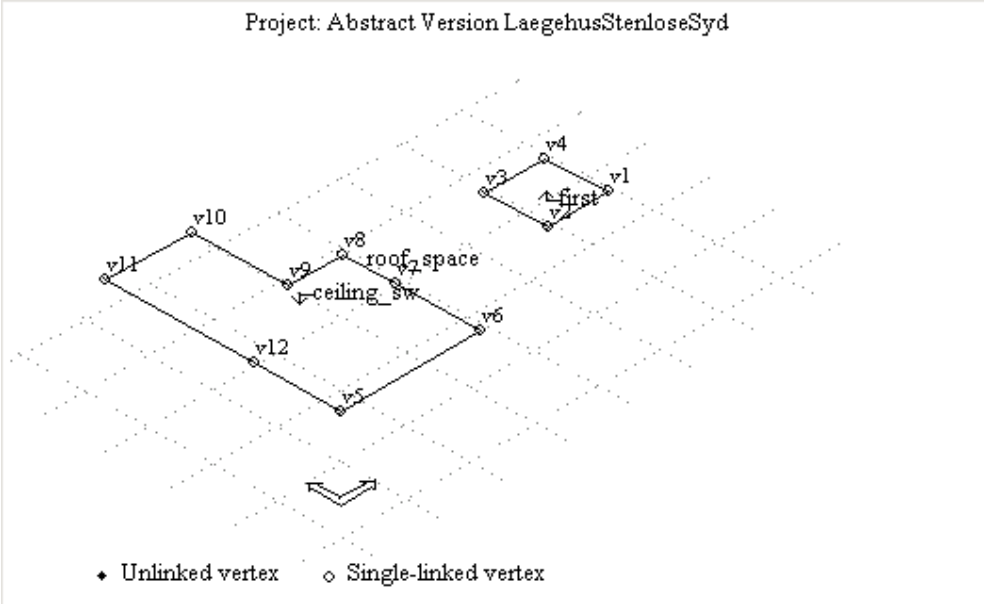


Figure 4.11: After import and inversion of a surface.

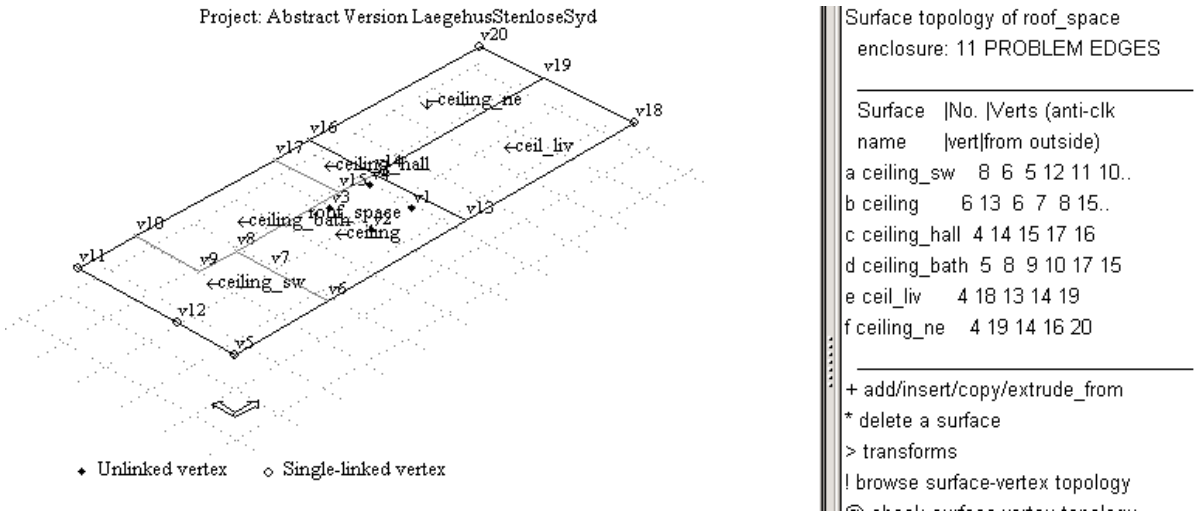


Figure 4.12: roof_space with imported ceilings.

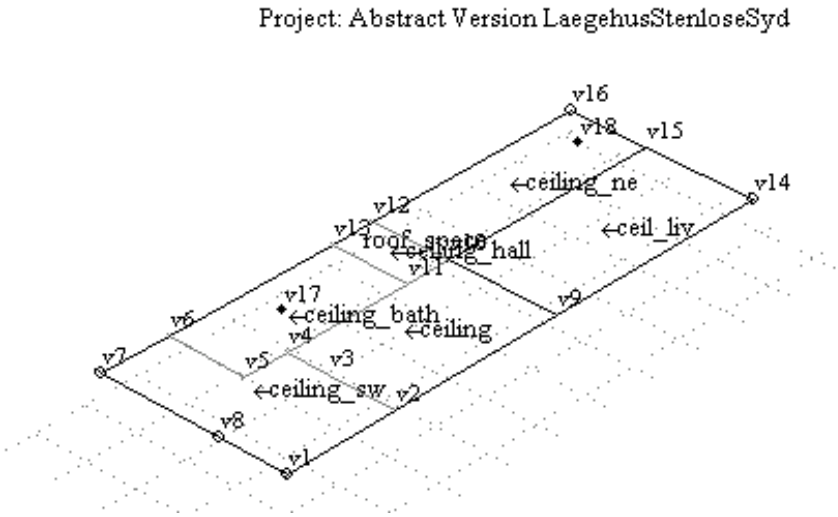


Figure 4.13: roof_space with two ridge vertices.

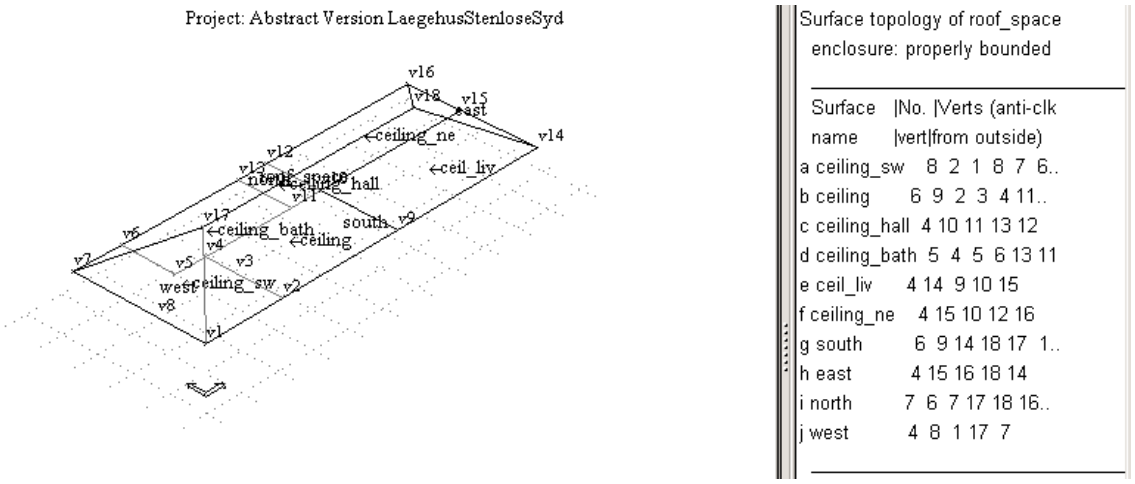


Figure 4.14: Zone with roof surfaces.

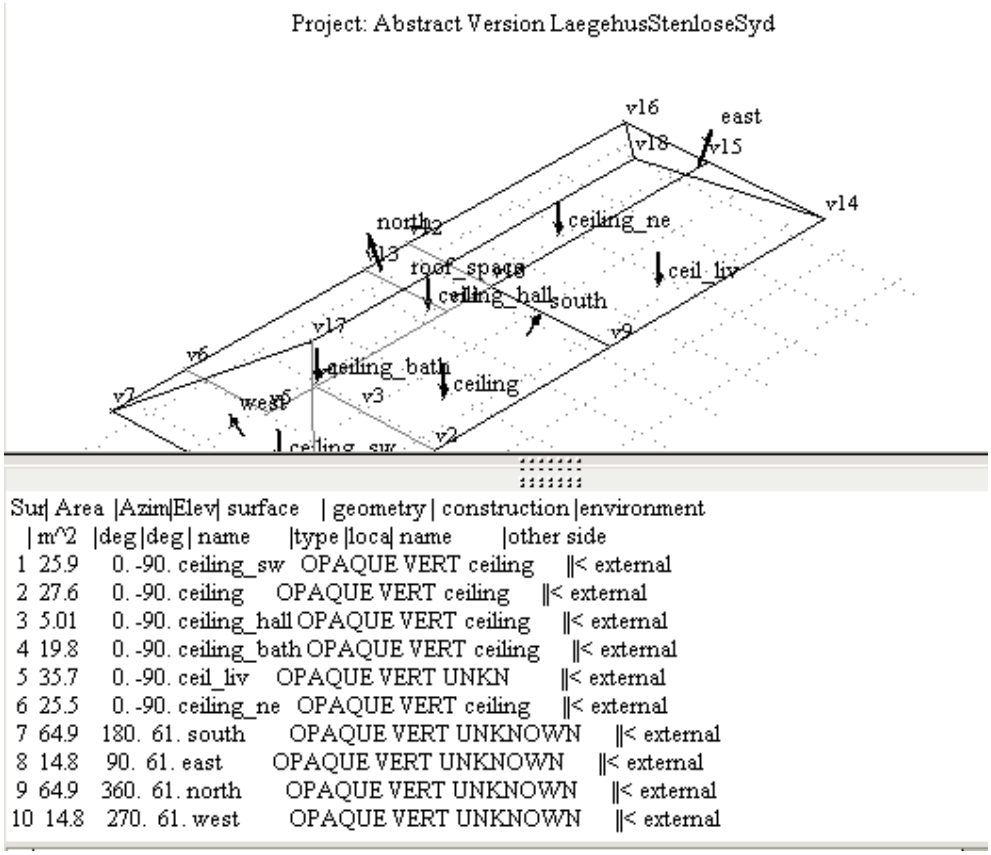


Figure 4.15: Better QA via combined wireframe and text report.

When you select a surface in another zone you will be asked if there are any transforms to apply. The key transform is 'invert' which takes the polygon defined in the other zone and reverses the order of the edges so that it faces the correct way within the roof_space.

For each surface being copied, the wire frame view is updated to show the other zone as an aide to selecting the correct surface. It helps if the surfaces in such lists are clearly named. Remember to select the 'invert' option as seen in Figure 4.15. After the first copy the roof space will look like

Figure 4.10.

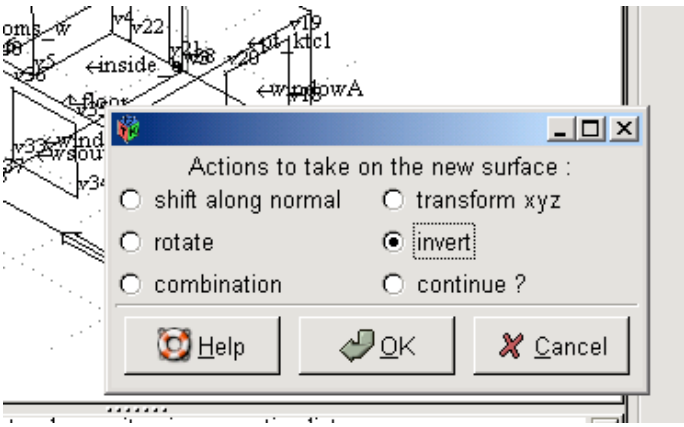


Figure 4.15 The invert dialogue.

During the import you might find a source surface with a duplicate name. You will be asked to specify a new unique name. Tactical hint: if you follow a clear naming strategy, subsequent tasks will go faster. At some point remember to remove the initial dummy surface. When all of the ceilings have been imported the roof_space will look like Figure 4.11. Each surface name provides a clue as to what is on the other side of the ceiling. Typically, each imported surface will require between 5 and 10 seconds for an experienced user and if there are name clashes it might require 20 seconds per import.

There are a number of steps which you can take at this point which will prevent the propagation of errors. For example, the wire frame image will have open circles drawn at each vertex that is referenced once - so open circles are to be expected at the borders. The wire frame image will have solid circles at vertices not referenced by any surface. You will see four such circles in the wire frame - these are orphan vertices associated with the initial dummy surface. It will save time and limit confusion if you exit from the `Surface topology` menu and go into the `vertex coordinates` menu and identify vertex 1, 2, 3 and 4 for removal.

While you are in the `Vertices in` menu take a note of the Z values. The current position of the surfaces is at 2.35m. We want the ridge of the roof to be 2m above this point. The roof is a hip type and the dimension from south to north is 7.2m so the so the ridge will start 3.6m from the East and West and South edge of the building facade. These two coordinates are then:

- left ridge point X=3.6, Y=3.6, Z=4.35
- right ridge point X=15.77, Y=3.6, Z=4.35

For a gable roof the X coordinate would not be altered to form a pair of triangular walls. The next step is to '+ add' two more vertices to the roof_space zone. The result should look like Figure 4.13 (look for v17 and v18).

The next step is to compose the South, East, North and West surfaces of the roof by creating new surfaces made up of existing vertices. With reference to Figure 4.14

- south surface should include vertices 9 14 18 17 1 and 2.
- east surface should include vertices 15 16 18 and 14.
- north surface should include vertices 6 7 17 18 16 12 and 13.

- west surface should include vertices 8 1 17.

There are patterns in the above definitions. What are they?

- Each surface is defined in an anti-clockwise order (looking from the outside)
- the first edge is horizontal and the second edge is not horizontal
- intermediate vertices (e.g. 2 and 9 and 15 etc.) are included in the list

The first edge horizontal rule is required by the shading and insolation analysis. Look at the surfaces in any of the simple exemplar models and you will see this pattern. With simple shaped surfaces there is usually only one edge along the base of a wall and for such surfaces the normal rule of 'start from the lower left edge going anti-clockwise' applies. In this case because there are several edges in sequence so the rule has to be adapted. If you miss out one of the intermediate vertices (e.g. vertex 15 along the east faade) ESP-r will detect a mismatch and warn you to check the polygon edges.

If you followed these steps, the interface should look like Figure 4.14. An experienced user will require about 10 seconds to create a new surface from existing vertices and will also be double checking that data glitches are caught early and corrected before progressing to the next task. Practice until you feel confident with the technique!

Notice the `enclosure: properly bounded` message at the top of the menu. This signals that all of the edges in the zone are following the rules of syntax and order and that the zone if fully bounded by polygons.

A further check could be done by turning on the surface normal arrows in the wire frame drawing (in the X11 interface this is found in the 'wire-frame' button and in the GTK interface the option is within the pop-up dialogue (as in Figure 4.5).

Almost finised. Now is a goot time to contrast the visual information in the wire frame with the `zone & surface details` report (see Figure 4.14). The few seconds that it takes to generate this report and review it with the wire frame will typically save tens of minutes later.

Having created the polygons and given them names, the next task is to attribute the surface composition. A copied surface already has inherited attributesd. The new surfaces of the roof are partially attributed at this point.

Remember that there is a automated process which looks at the co-ordinates of each surface to find thermophysical adjacency. It may be quicker than attributing each surface in the roof_space manually. Generate a QA report and check the contents before you continue.

4.3 Shading obstructions

The roof is formed and attributed but the walls will not be shaded until obstruction blocks are added to the model. This section is incomplete.

Other attribution such as casual gain and air flow schedules are covered elsewhere.

5 Schedules

The form and composition of a model is one part of the simulation process. Many users think they have almost completed their work when the geometry is done. Far from it, buildings are almost always places where people are coming and going and lights are being turned on and off and all manner of electrical devices are found.

Usually we lack both the detailed information and the resources to undertake an exhaustive definition. It is, however, in our interest to define the essential characteristics of what goes on in a building and learn from the performance patterns that emerge sufficient clues to imagine the circumstances where the building would perform poorly.

ESP-r supports zone operational characteristics in terms of weekdays and two separate weekend days (typically labeled as Saturday and Sunday). Work is underway to implement more day types but for this exercise lets stick with the basics. It is also possible to define unique values for each timestep of a simulation, but lets not go there yet. Casual gains (e.g. people, light, small power) are one operational characteristic of a zone and schedules of infiltration (air from the outside via intentional sources such as fans or unintentional sources such as cracks in the facade) and ventilation (air from another thermal zone) and there are a limited number of controls you can impose on infiltration and ventilation schedules.

What are casual gains in ESP-r? The lower portions of Figures 1.9 and 1.10 show the attributes of the casual gains - each has a day type (Wkd/Sat/Sun) a casual gain type label (Occupt/Lights/Equipt) a period (start hour and end hour) a sensible and latent magnitude, and for the sensible portion the fraction of the gain which is radiant and the fraction which is convective. The convective and radiant fraction defaults should be adjusted to reflect the properties of specific light fittings and occupants.

Just before creating a schedule be aware that groups who frequently work with a particular building type will likely have historical data as well as prior models which could contain useful patterns of occupancy and small power use. There are several techniques for helping to re-use such information and this will be covered later in this section.

To give you more practice with creating zone schedule also have a look at Exercise 13 in the *Cookbook Exercises* volume.

For the current building a brief description was given in the 'How the building

is used' section, specifically Figure 1.10 and 1.11. Re-read that section. Also look at Figure 53 for reception data. Before we define the operational characteristics of the reception and examination room within the Project Manager a bit of planning will (you guessed it) save time and reduce the chance of errors later on.

In the reception there is 'one staff and up to 3 visitors with 10W/m2 lighting and 1 50W' From the figure it is clear the occupancy changes throughout the day (ramping up from 7h00 and with a dip for lunch and almost nothing happening after 17h00) but the lights are on during office hours plus some time for cleaning staff (8h00-19h00). In the examination room there is one staff and one visitor with 10W/m2 lighting and 1 100W computer '. From the figure it is clear that occupancy varies during the day and that both lights and small power (labeled as equipment) are on from 8h00-19h00 and nothing happens on the weekend.

Why bother with varying the occupancy during the day? Several reasons - full time peak loads usually do not happen in reality so a bit of diversity is more realistic, reducing the load during a lunch hour allows us to check whether the building is sensitive to brief changes in gains and the ramp-up just before office hours and the ramp-down after office hours approximates transient occupancy. The peak demands are long enough to indicate whether heat will tend to build up in the rooms. Such patterns will also exercise the environmental system and perhaps provide an early clue as to the relationship between building use and system demands. Quite a lot of value for a few minutes thought at the planning stage.

In the reception the peak occupant sensible gain is 400W and in the examination room the peak is 200W equating to 100W per person. The latent magnitude is roughly half the sensible value. Such assumptions, if documented, really do help clarify the numbers held in the file and can speed up later QA tasks.

One of the first questions you will be asked when you begin to define the operational characteristics of zones is the number of periods for each day type and the start time of each period. For weekdays the reception has 8 occupant periods and 3 periods for lighting and 3 periods for small power. According to Figure 1.9 the occupant periods would start at 0, 7, 8, 9, 12, 14, 17 and 21 hours. On Saturday and Sunday there are zero periods.

In the Project Manager go to `Model Management - browse/edit/simulate -> composition -> operational details.` Select the zone reception and you will be presented with an initial file choice with an option to browse for an existing file within your model (you could use this if you have several rooms which use the same pattern). The file name is suggested based on the name of the zone. Accept the suggested name and then you are presented with a number of options for how to define the schedules (Figure 5.1).

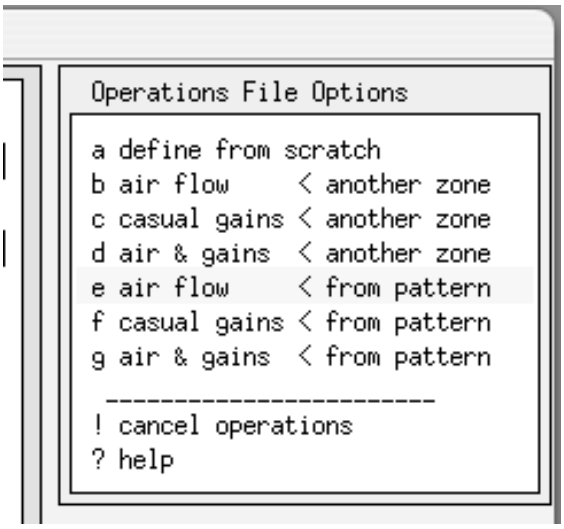


Figure 5.1 Options for generating schedules.

For this exercise we will use the `start from scrach` option. You can also import patterns from other existing zones in your model as well as from zone operation files that have been placed in a standard 'pattern' folder.

You will be reminded about planning your schedules (do read this because it is a useful reminder). And then for each of the day types you will be asked for the number of periods for occupants, lights and small power. This will be followed by dialogues which ask for the start hour of each of the periods for each of the casual gain types (get this from your notes).

Do the same for for Saturdays and Sundays. You are now presented with the menu in Figure 5.2.

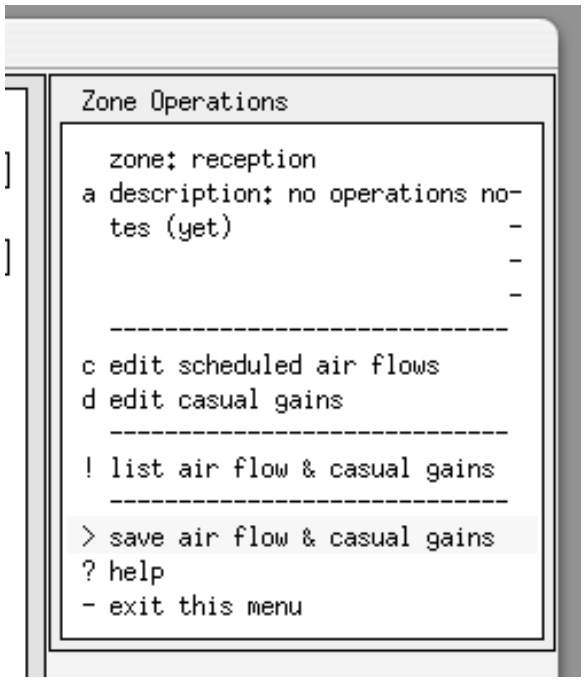


Figure 5.2 Opening zone operations menu.

Fill in the description of the zone operations using words and phrases which will clarify what is happening (note the X11 editing box has < > arrows so you can scroll to a more text). Next select option c to fill in the rest of the casual gain period details. You will be presented with a menu with period data which you need to fill in based on your notes.

As you were filling in the period data you will note that you have a choice of units. From the notes occupants are Watts, lighting would be 3.75 W/m2 if we used that unit (the notes say 150W and the floor area is 40m2), and small power as Watts.

After you have defined the magnitude of the sensible and latent gains and accepted the default radiant and convective split you should see something like Figure 5.3.

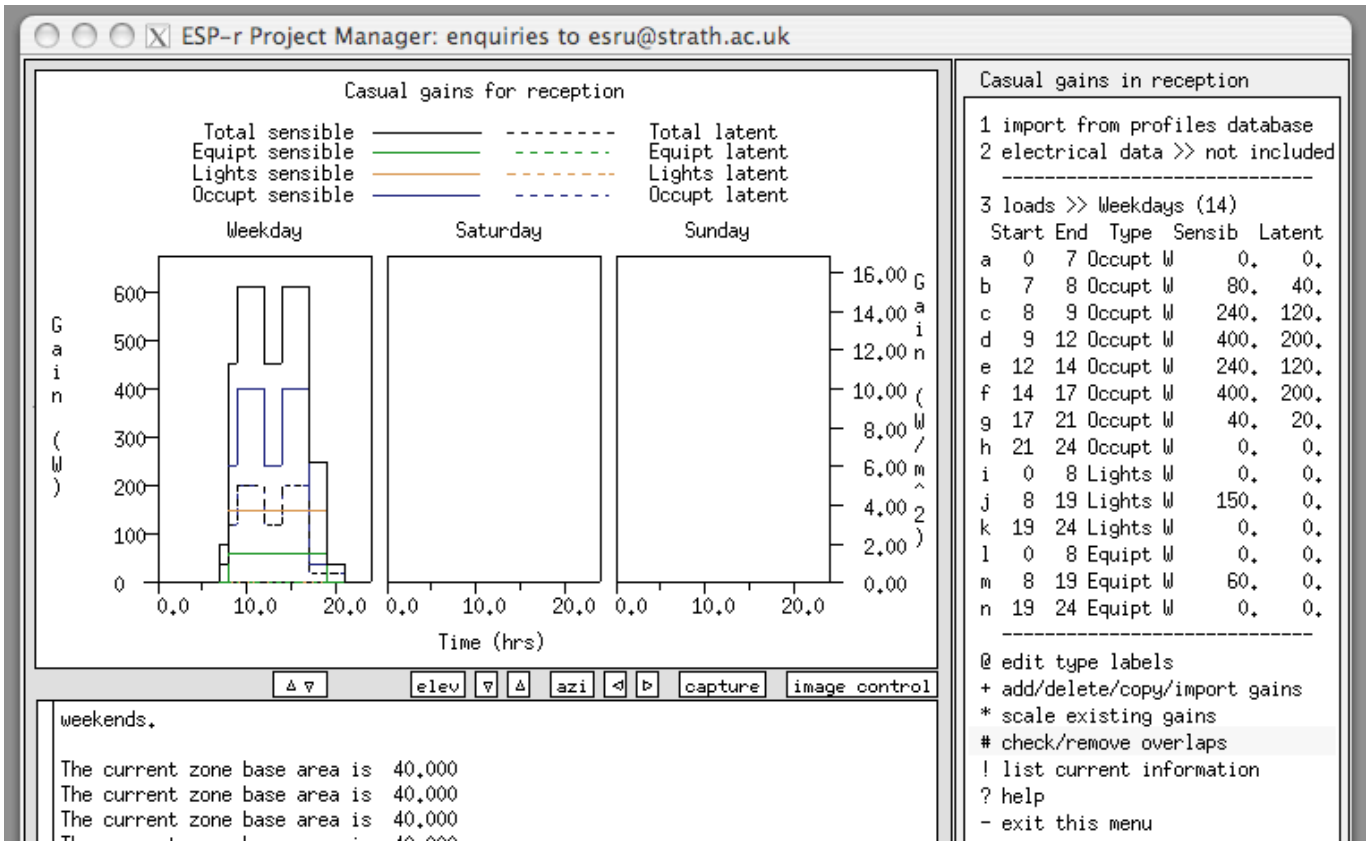


Figure 5.3 Weekday casual gains.

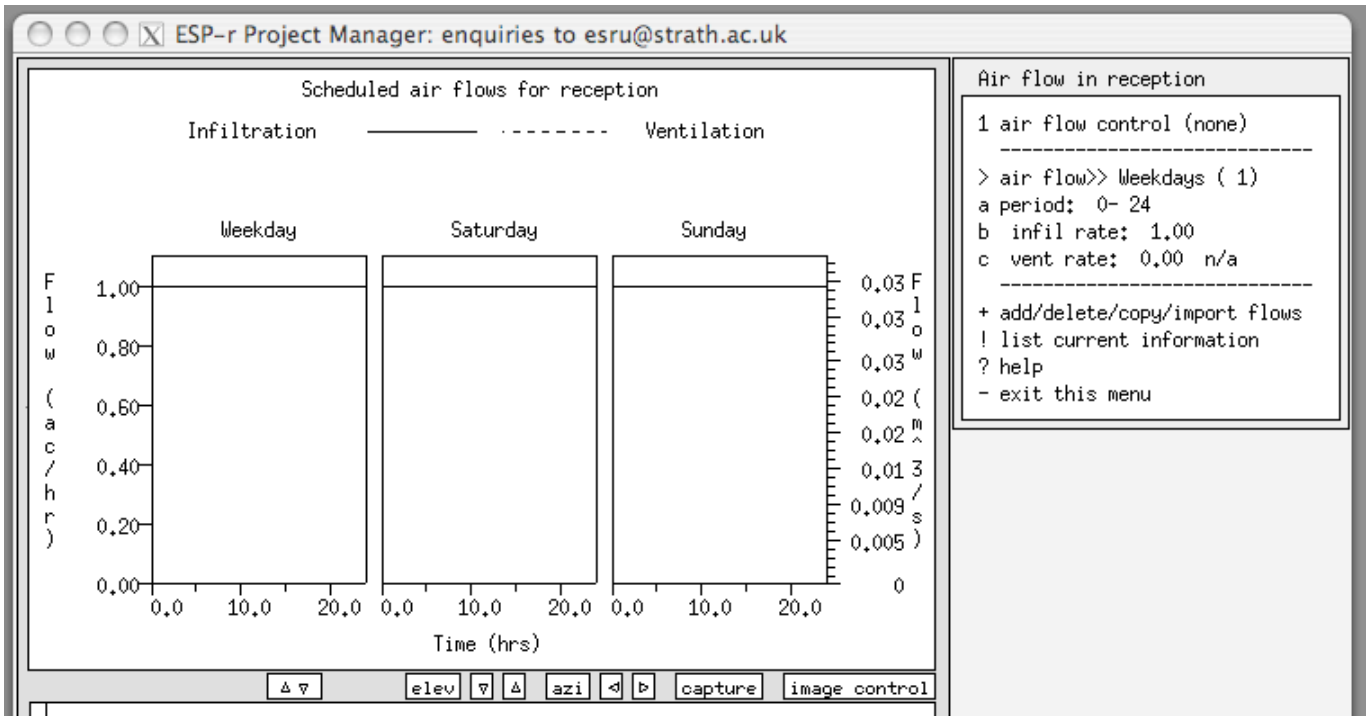


Figure 5.4 Weekday infiltration schedules.

Using the information on your notes you could also define the casual gains for the examination room.

5.1 Scheduled air flows

Early in the design process building details may not support a detailed description of how air moves within buildings or how tight the facade might be so engineering approximations are often used. The brief didn't actually mention anything so an initial assumption needs to be made. In an actual project discussions would be made within the design team to quantify this figure. For purposes of this exercise lets have a rather leaky facade and assume that the doors are closed between the reception and examination room.

We can represent this with one period each day covering the 24 hours with a value of 1 ac/h infiltration and no ventilation (see Figure 5.4). Use the `add/delete/copy import flow` option and select `add` for all day types. Later we might decide to lower the infiltration rate to see if the building is sensitive to an upgrade in the quality of the facade.

Other sections of the *Cookbook* outline options for treating air movement via mass flow networks which can assess pressure and buoyancy driven flows between thermal zones or via Computational Fluid Dynamic Domains.

5.2 Importing operation schedules

Creating zone operations from scratch is time consuming so many users will collect their best zone operation patterns and store them in the pattern folder (located with the training models).

If you have already filled in the schedules for the examination room repeat the process but change the name of the file slightly so as not to mess up your prior work. When the selections in Figure 5.1 are shown pick `air and gains < from pattern` and a list of files will be shown. Select one of them (remember which one) and look at the summary and answer the questions about air flow and then about casual gains. Since it does not know the volume or the floor area associated with the pattern file it needs information from you. If the author of the file you are getting the pattern from was really clued-up such information would be included in the documentation.

You will have a opportunity to edit the documentation and this is you chance to ensure that what is included in your model is clearly defined, especially if you need to scale some of the values. Figure 5.5 is the result of importing a pattern file. Such patterns can be altered easily - note that the peak value for occupants and lighting and small power are each 100W. To upscale the small power for use in your current zone would require only that you select the `scale existing gains` option and provide a scaling factor.

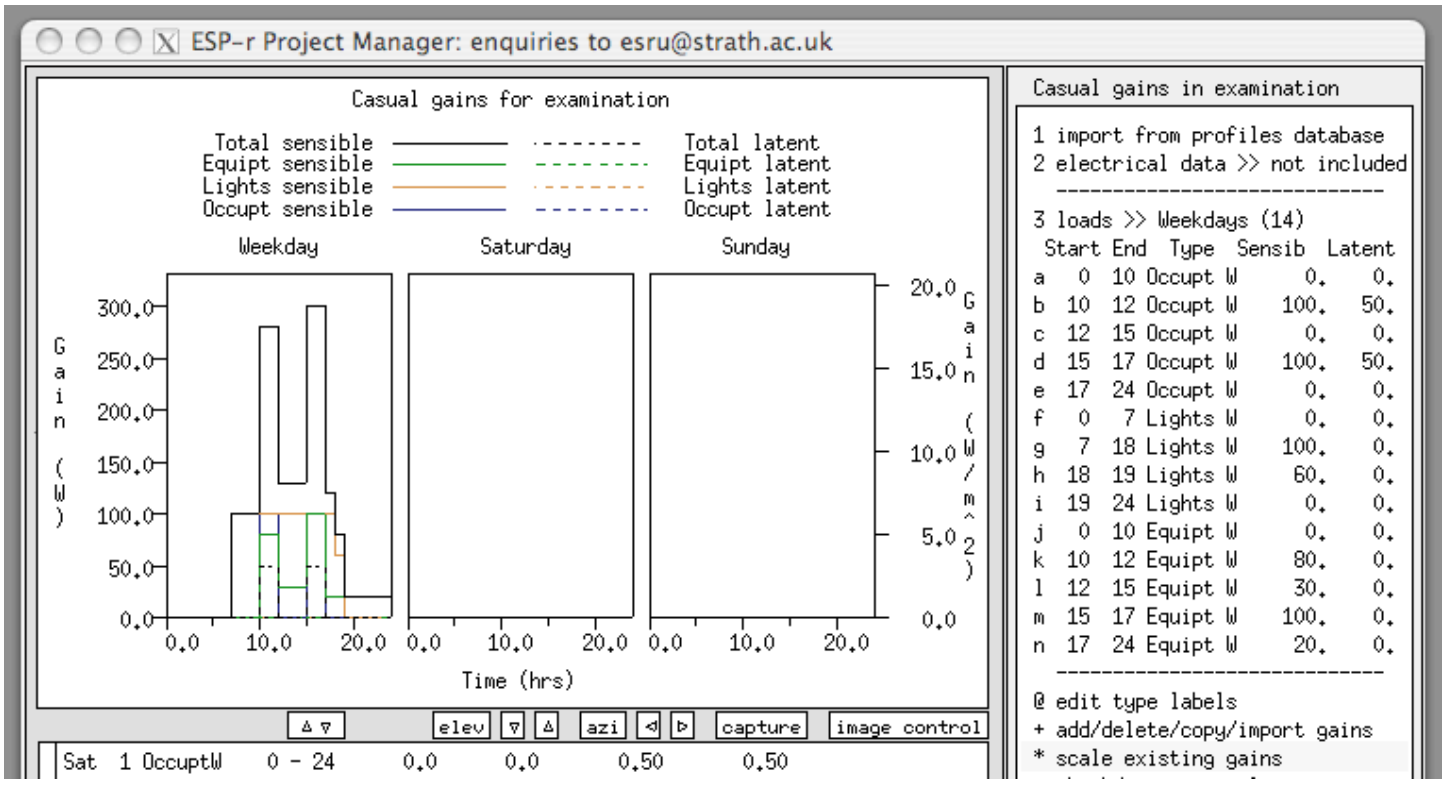


Figure 5.5 Imported pattern file data.

Once a model includes the zone geometry and the thermophysical data files and the schedules of use it is possible to run a simulation. If you think you are at this point then have a look at Exercise 15 in the *Cookbook Exercises* volume and see if you can assess your models performance. If you have not defined environmental systems then the assessment will be based on a free float assumption.

6 Climate data

Unless we find ways to focus and constrain our models and the assessments we carry out, simulation tasks risk demanding more computing and staff resources then we have available. In this section the focus one one way to deliver information faster (and without exhausting ourselves) by resisting our impulse to run long assessments early in the design process.

ESP-r has a number of facilities which allow us to scale our models and assessments - so we get, for example, annual heating and cooling for a whole building without simulating every day of the year or every floor of an office building.

For many building types there is a strong correlation between predictions over one or two weeks in each season and seasonal demands. If we can establish that this is the case for a building, scaling of performance predictions becomes an option.

The ESP-r climate data sets can be supplemented with information about seasons (early-year winter, spring, summer, autumn, late-year winter in the Northern Hemisphere, early-year summer, autumn, winter, spring later-year summer in the Southern Hemisphere) as well as typical assessment periods.

Seasons are partly based on local conventions and partly on science. We possible the demarcation between seasons is based on input from the simulation community. There are several approaches to deriving a typical assessment period from the climate data file and then placing it where it is easy to access.

6.1 Importing climate data

To better understand how this works our first task is to install a new climate file and specify the days in each season and then use clm facilities to discover typical assessment periods in each season. After this we will derive scaling factors for heating, cooling, lighting, small power etc. demands to use in our model.

Downloading a new climate data set from a United States DoE web site (on one line)
`<http://www.eere.energy.gov/buildings/energyplus/cfm/weatherdata.cfm>.` The site offers US locations, Canadian Locations and International Location. Lets choose an international location - Geneva Switzerland. The file for this site is CHE_Geneva_IWEC.zip. Download it and save to a convenient location and unpack the zip file. One of the files will be CHE_Geneva_IWEC.epw. Most current EPW files can be imported directly into the ESP-r clm

module.

For Linux/Mac/Unix use a command window, go to the location of the EPW file and issue the command (as one line): `clm -mode text -file CHE_Geneva_IWEC -act epw2bin silent CHE_Geneva_IWEC.epw.`

This creates a new ESP-r climate file CHE_Geneva_IWEC. The message `error reading line 1` is sometimes seen when doing the conversion, but does not usually affect the conversion. The command given to the clm module includes the name of the new binary climate file to be created after the key word -file. The words `-act epw2bin silent` tells it to undertake the conversion without further interactions.

For the Native Windows version you will have to start up the clm.exe module interactively and set the new ESP-r climate file and and provide the name of the EPW file and use the `import` option.

To check that the conversion worked, invoke the clm module with the new file or use the file browser to locate the new file:

```
clm -file CHE_Geneva_IWEC
```

If the conversion was correct you should see the lines

```
Climate data: GENEVA CHE
46.2N  8.9W: 1984  DN
```

in the text feedback area of the interface. And for good measure try to graph temperatures and solar radiation.

If there is an issue with the file then open it in a text editor (nedit is used in Figure 6.1). There are several changes we might need to make in the file.

Line 6 include a # character before the WMD number and this should be replaced by a blank character. Line 6 is also 702 characters long and ESP-r can only read text lines that are less than 248 characters. After reading the ASHRAE notice, edit the notice text shorten it. The last change is that some EPW files have a blank line at the end of the file (line 8769). Remove this line and save the file. Further instructions for working with EPW files will be found in the ESP-r source distribution in the climate folder.

6.2 Defining seasons and typical periods

Our next task is to define the days associated with each season. There are any number of approaches one might take and we will use a combination of looking at the patterns of temperatures over the year and the solar radiation. In the climate module choose graphical analysis from the menu options. Then pick dry bulb temperature and draw graph to get a display similar to that in Figure 6.2.

CHE_Geneva_IWEC.epw – /Users/jon/Documents/cookbook/

File Edit Search Preferences Shell Macro Windows Help

/Users/jon/Documents/cookbook/CHE_Geneva_IWEC.epw DOS byte 0 of 1412272 L: 1 C: 0

```
LOCATION,GENEVA,-,CHE,IWEC Data,067000,46.25,6.13,1.0,416.0
DESIGN CONDITIONS,1,Climate Design Data 2005 ASHRAE Handbook,,Heating,1,-7,-4.9,-11.1,1.5,-4.9,-9,1.8,-2.8,10.4,2.
TYPICAL/EXTREME PERIODS,6,Summer - Week Nearest Max Temperature For Period,Extreme,7/ 6,7/12,Summer - Week Nearest
GROUND TEMPERATURES,3,.5,,,,2.79,2.05,3.37,5.43,10.84,15.02,17.83,18.69,17.23,13.99,9.66,5.64,2,,,,5.53,4.20,4.48,
HOLIDAYS/DAYLIGHT SAVINGS,No,0,0,0
COMMENTS 1,"IWEC- WMO#067000 - Europe -- Original Source Data (c) 2001 American Society of Heating, Refrigerating
COMMENTS 2, -- Ground temps produced with a standard soil diffusivity of 2.3225760E-03 {m**2/day}
DATA PERIODS,1,1,Data,Sunday, 1/ 1,12/31
1984,1,1,1,60,C9C9C9C9*0?9?9?9?9?9?9?9A7A7A7A7A7A7*0E8*0*0,-2.4,-4.9,81,98100,0,1415,236,0,0,0,0,0,0,0,250,1.0,0,0
1984,1,1,2,60,C9C9C9C9*0?9?9?9?9?9?9?9*0B8B8B8B8*0*0E8*0*0,-1.8,-3.7,85,98100,0,1415,239,0,0,0,0,0,0,0,250,0.8,0,0
1984,1,1,3,60,C9C9C9C9*0?9?9?9?9?9?9?9*0B8B8B8B8*0*0E8*0*0,-1.4,-2.9,88,98100,0,1415,241,0,0,0,0,0,0,0,250,0.7,0,0
1984,1,1,4,60,C9C9C9C9*0?9?9?9?9?9?9?9A7A7A7A7A7A7*0E8*0*0,-1.4,-2.6,90,98100,0,1415,241,0,0,0,0,0,0,0,240,0.5,0,0
1984,1,1,5,60,C9C9C9C9*0?9?9?9?9?9?9?9*0B8B8B8B8*0*0E8*0*0,-1.5,-2.6,91,98100,0,1415,241,0,0,0,0,0,0,0,240,1.2,0,0
1984,1,1,6,60,C9C9C9C9*0?9?9?9?9?9?9?9*0B8B8B8B8*0*0E8*0*0,-1.7,-2.6,93,98100,0,1415,240,0,0,0,0,0,0,0,240,1.9,0,0
1984,1,1,7,60,A7A7E8E8*0?9?9?9?9?9?9?9A7A7A7A7A7A7*0E8A7*0,-1.8,-3.0,90,98100,0,1415,239,0,0,0,0,0,0,0,240,2.6,0,0
1984,1,1,8,60,B8C8E8B8*0?9?9?9?9?9?9?9*0B8B8B8B8*0*0E8*0*0,-1.4,-2.6,90,98100,0,1415,241,0,0,0,0,0,0,0,240,1.7,0,0
1984,1,1,9,60,B8C8E8B8*0H9H9H9I9I9I9I9*0B8B8B8B8*0*0E8*0*0,-1.0,-2.3,90,98100,37,1415,243,11,0,11,1200,0,1200,340,
1984,1,1,10,60,A7A7E8E8*0G9G9G9I9I9I9I9A7A7A7A7A7A7*0E8*0*0,-0.6,-2.0,89,98100,211,1415,244,89,53,81,9700,3800,900
1984,1,1,11,60,B8C8E8B8*0H9H9H9I9I9I9I9*0B8B8B8B8*0*0E8*0*0,0.1,-1.3,89,98000,360,1415,258,186,361,94,19400,28800,
1984,1,1,12,60,B8C8E8B8*0H9H9H9I9I9I9I9*0B8B8B8B8*0*0E8*0*0,0.9,-0.4,91,98000,458,1415,272,197,130,155,21400,11500
```

Figure 6.1 Editing of EPW file.

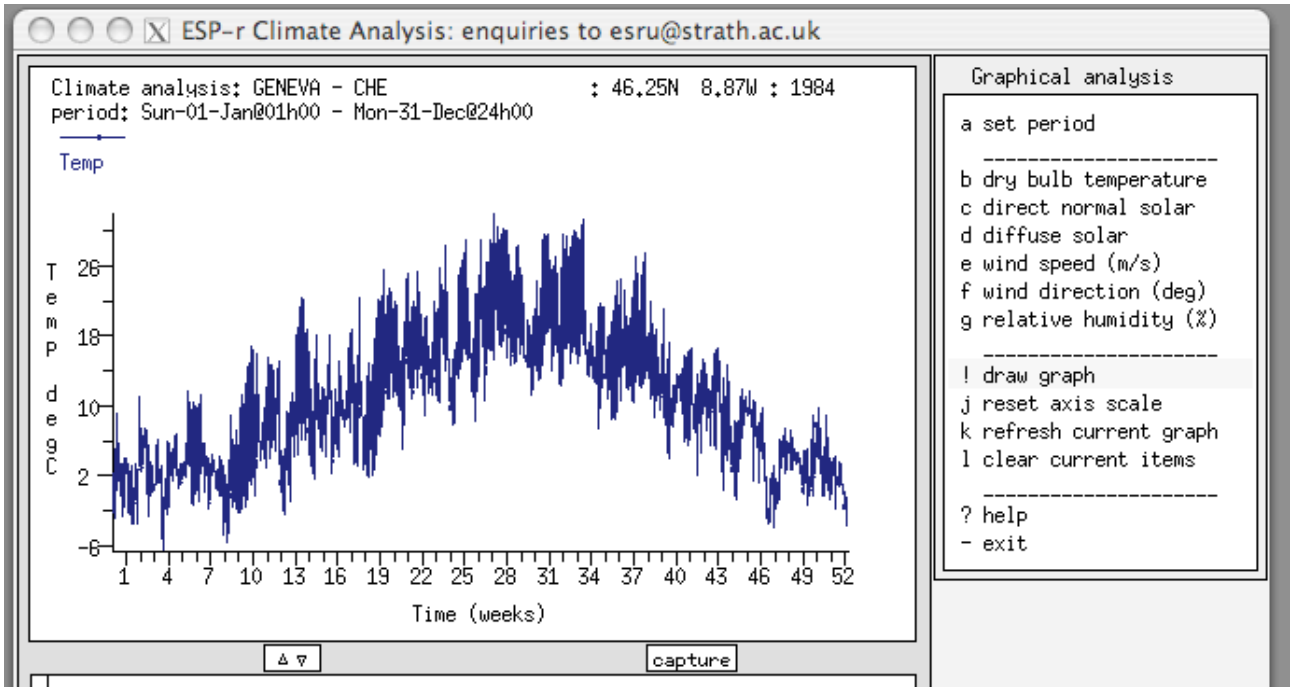


Figure 6.2 Annual bry bulb temperatures.

The axis at the bottom of the graph is weeks. There are extreme low temperatures in weeks 4, 9, 46 and 52. There is a late cool period in week 12. In week 14 it reaches 22°C. The warmest period is between week 26 and week 32.

Another way to look at climate information is to look at weekly heating and cooling degree days. To to this select synoptic analysis and then choose dry bulb temperature and then degree days and then weekly. Take the default base heating and cooling temperatures This will produce a table as Figure 6.3.

```
Climate data: GENEVA - CHE
   46.2N    8.9W: 1984  DN
Degree day analysis: heating base at  17.0 & cooling  21.0 Deg C
Week    (starting)       Heat dd          Cool dd
                         avg/day  total   avg/day  total
Week: 1 (Sun-01-Jan)     15.43   108.02    0.00     0.00
Week: 2 (Sun-08-Jan)     15.85   110.93    0.00     0.00
Week: 3 (Sun-15-Jan)     14.01    98.08    0.00     0.00
Week: 4 (Sun-22-Jan)     15.80   110.60    0.00     0.00
Week: 5 (Sun-29-Jan)     14.29   100.02    0.00     0.00
Week: 6 (Sun-05-Feb)     13.02    91.16    0.00     0.00
Week: 7 (Sun-12-Feb)     13.93    97.52    0.00     0.00
Week: 8 (Sun-19-Feb)     15.83   110.82    0.00     0.00
Week: 9 (Sun-26-Feb)     15.90    95.42    0.00     0.00
Week:10 (Sun-04-Mar)     12.75    89.26    0.00     0.00
Week:11 (Sun-11-Mar)      9.83    68.81    0.00     0.00
Week:12 (Sun-18-Mar)      8.05    56.34    0.00     0.00
Week:13 (Sun-25-Mar)     12.30    86.08    0.00     0.00
Week:14 (Sun-01-Apr)      5.25    36.78    0.03     0.18
Week:15 (Sun-08-Apr)      7.94    55.60    0.00     0.00
Week:16 (Sun-15-Apr)      8.18    57.28    0.00     0.00
Week:17 (Sun-22-Apr)      7.21    50.46    0.00     0.00
Week:18 (Sun-29-Apr)      6.53    45.73    0.01     0.08
Week    (starting)       Heat dd          Cool dd
                         avg/day  total   avg/day  total
Week:19 (Sun-06-May)      8.25    57.75    0.02     0.13
Week:20 (Sun-13-May)      2.53    17.73    0.31     2.15
Week:21 (Sun-20-May)      3.30    23.07    0.12     0.84
Week:22 (Sun-27-May)      1.16     8.15    0.31     2.18
Week:23 (Sun-03-Jun)      2.70    18.89    0.01     0.08
Week:24 (Sun-10-Jun)      0.95     6.66    0.69     4.81
Week:25 (Sun-17-Jun)      3.25    22.75    0.63     4.38
Week:26 (Sun-24-Jun)      1.55    10.87    0.82     5.76
Week:27 (Sun-01-Jul)      0.63     4.39    1.45    10.16
Week:28 (Sun-08-Jul)      0.03     0.24    3.33    23.33
Week:29 (Sun-15-Jul)      0.16     1.14    1.70    11.90
Week:30 (Sun-22-Jul)      1.08     7.55    0.32     2.27
Week:31 (Sun-29-Jul)      0.82     5.71    1.60    11.17
Week:32 (Sun-05-Aug)      0.39     2.75    1.58    11.05
```

Figure 6.3 Heating and cooling degree days.

The average heating degree days is roughly 15.0 for the first nine weeks and then drops to roughly 8.0 (except for week 13). Weeks 27 to 34 have cooling degree days between 10 and 23. This follows the same pattern as seen in the graph. If we set a winter heating degree day cutoff point of 12 and a summer cooling degree day cut-off of about 10 then the definition of seasons is straightforward.

Before we actually set the dates, note that 1 January is on a Sunday and the start of each subsequent week is also a Sunday. Later, when we search for typical weeks they will begin on a Sunday. Some practitioners prefer to run assessments that begin on Mondays and end on Sundays. If we wanted to enforce that preference what we need to do is to change the year of the climate set so that January happens on a Monday (e.g. 2001). This change can be found in the `edit site data` menu option of the main clm menu. Once this is changed, return to synoptic analysis and ask for the weekly degree day table again.

Using these cutoffs the seasons are as:

- early-year winter 1 January - 11 March
- spring 12 March - 24 June
- summer 25 June - 26 August
- autumn 27 August - 18 November
- late-year winter 19 November - 31 December.

To record this information go to the `manage climatelist` option on the main menu. You will be presented with the options shown in Figure 6.4.

What is shown are initial default dates for seasons and typical periods which you will need to update. If the `menu string` item and the `menu aid` are not clear then start by editing the menu selection and documentation text.

For example the menu aide memoire could be Geneva CHE was source from US DoE. Menu option c is the full path and name of the climate file that ESP-r will access *after it has been installed in the standard location*. For the file we have been working with this is /usr/esru/esp-r/climate/CHE_Geneva_IWEC. Menu option d is a toggle which tells ESP-r whether the file is ONLINE or OFFLINE. Set this to ONLINE. If it is

OFFLINE then others will not see this file.

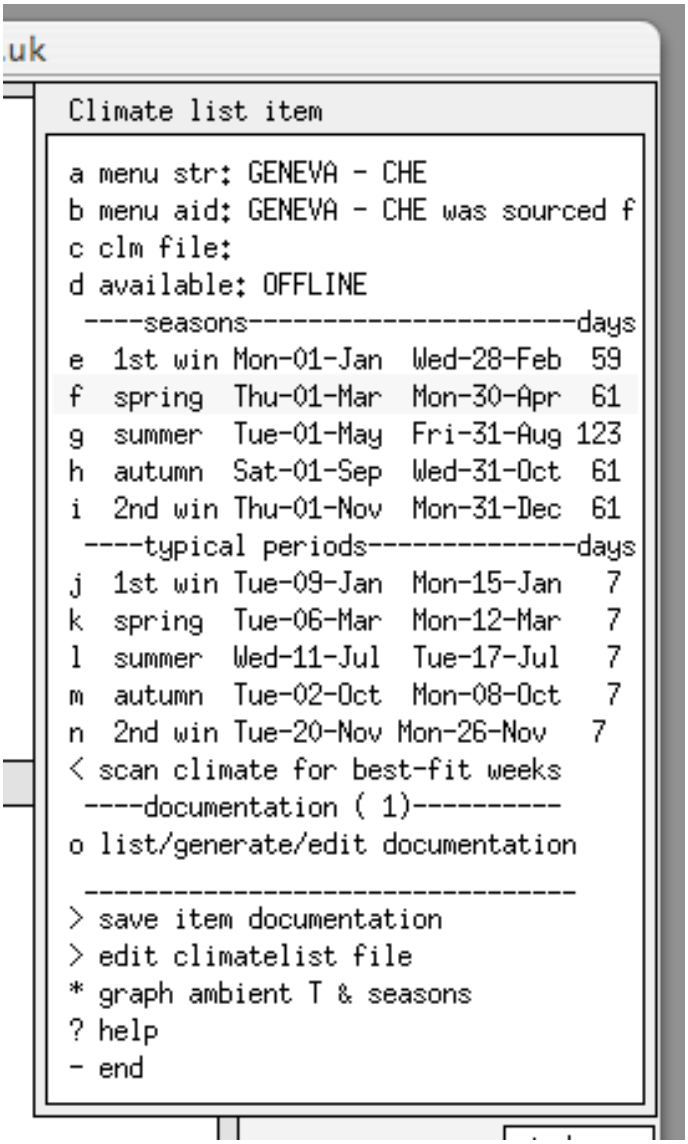


Figure 6.4 Menu for creating a *climatelist* entry.

The section of the menu under seasons allows us to edit the beginning and ending date of each season. After defining each season we can then use the scan climate for best-fit weeks to search for weeks that are closest to the seasonal average conditions. Notice that in this case, each season begins on the same day of the week. You could also define seasons that do not start at the same day of the week.

The criteria for heating and cooling are based on a combination of heating and cooling degree days and solar radiation. For example, the seasonal average weekly heating degree days and cooling degree days (102 and 0 for early year winter) as well as the solar radiation (11.05). It will then look for the week with the least deviation after confirming the weighting we want to give to heating DD, cooling DD and radiation. These are initial set to 1.0 to give an equal weighting (but you can change this if you want). It finds the smallest deviation (0.14) for the week eight which starts on Monday 19 February. This method can result in a good estimate of the energy use over a season although it will be less accurate for worst case peak assessments.

Use the `scan climate for best-bit weeks` option to search for the typical weeks. After confirming each of the seasonal suggestions and selecting the `graph ambient T and seasons` option the interface should look like Figure 6.5.

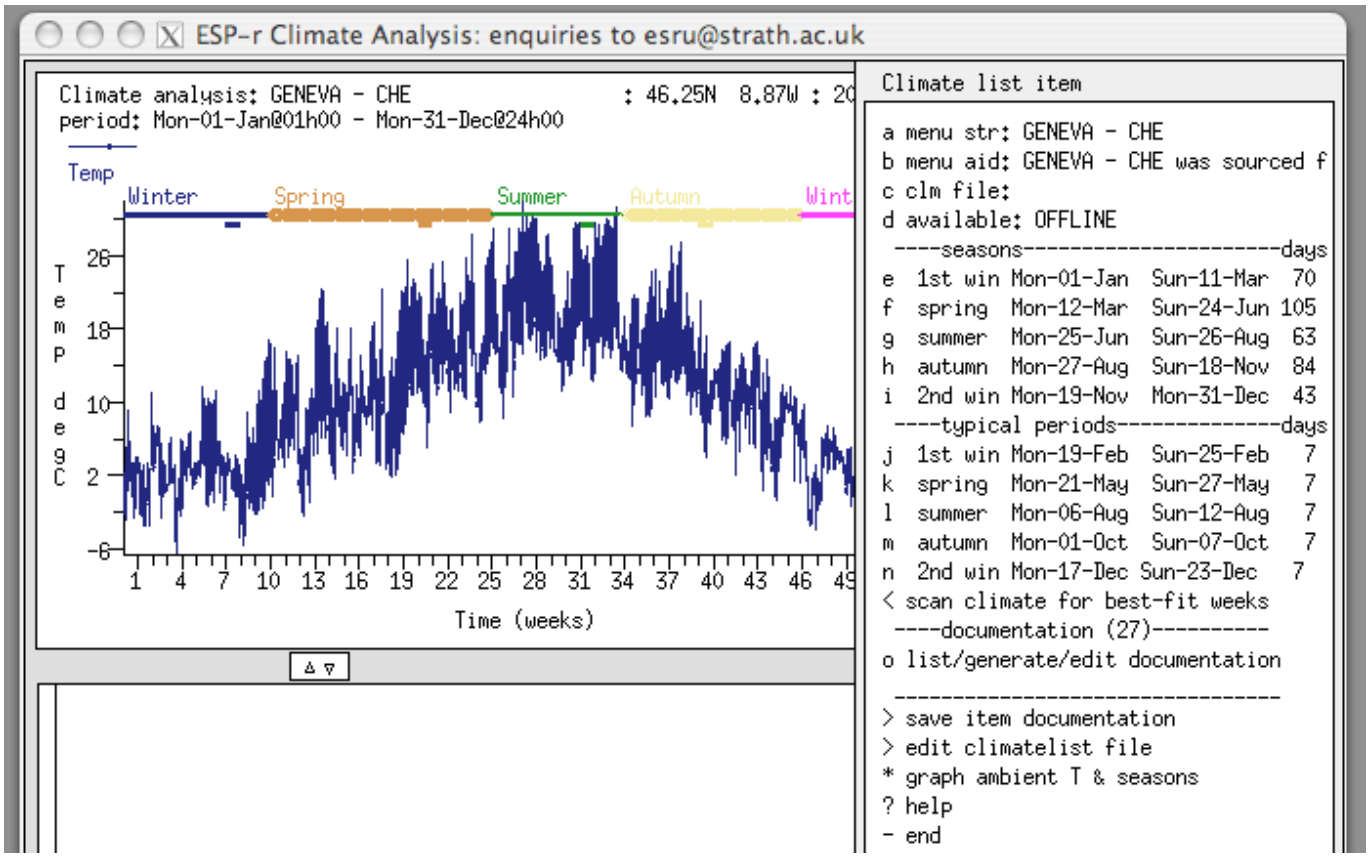


Figure 6.5 Seasons of the year.

6.3 Climatelist entries

The final tasks are to record this information via the `list/generate/edit documentation` option initialise option and then use the `save` option to write out the data to a file. It will give it a name based on the original climate file with a .block extension. The block of text that was generated is listed below. It needs to be pasted into the so-called *climatelist* file. There is an edit option. Also open the block file, check the entries and to add any text you want to have displayed to users (in the *help_start to *help_end).

Insert the text (carefully) between an existing *help_end and *item line and save it. Don't forget to copy your newly created climate file to the standard folder. The next time ESP-r is used the new climate file should be available and the seasons and typical weeks should be registered.

Before closing the clm module, it is useful to save the ESP-r climate data into an ASCII format file. Do this via the `export to text file` option of the main menu. Accept the default file name CHE_Geneva_IWEC.a and the period. The climate file CHE_Geneva_IWEC should be placed in the standard folder (e.g. /usr/esru/esp-r/climate) and the ASCII version CHE_Geneva_IWEC.a should be kept as a backup in case the binary climate file becomes corrupted. Again, on some systems, you may have to ask administrative staff to copy to file to /usr/esru/esp-r/climate (or wherever the file climatelist is located on your machine).

```
*item
*name GENEVA - CHE
*aide GENEVA - CHE was sourced from US DoE
*dbfl
*availOFFLINE
*winter_s  1  1 11  3 19 11 31 12
*spring_s 12  3 24  6 27  8 18 11
*summer_s 25  6 26  8
*winter_t 19  2 25  2 17 12 23 12
*spring_t 21  5 27  5  1 10  7 10
*summer_t  6  8 12  8
*help_start
Climate is GENEVA - CHE
 Location: 46.25N  8.87W : 2001
 Month  Minimum  Time      Maximum  Time         Mean
 Jan   -6.8 @ 4h00 Fri 26  11.1 @16h00 Sun 14  1.1
 Feb   -5.8 @ 7h00 Tue 27  11.7 @16h00 Wed  7  2.4
 Mar   -2.7 @ 7h00 Tue 27  16.8 @16h00 Sat 10  5.9
 Apr    1.4 @ 7h00 Wed 11  22.4 @16h00 Wed  4 10.1
 May    1.6 @ 4h00 Tue  8  25.5 @16h00 Tue 15 13.4
 Jun    7.7 @ 1h00 Tue 19  29.0 @16h00 Mon 25 16.7
 Jul   10.5 @ 4h00 Thu  5  32.1 @16h00 Mon  9 20.3
 Aug    7.1 @ 4h00 Thu 30  31.4 @13h00 Wed 22 19.9
 Sep    8.3 @ 7h00 Fri  7  27.6 @16h00 Sat 22 15.7
 Oct    0.1 @ 7h00 Wed 31  21.1 @13h00 Mon  1 10.6
 Nov   -4.1 @ 7h00 Sun 25  14.7 @16h00 Fri  2  4.7
 Dec  -4.0 @22h00 Mon 31    9.8 @13h00 Mon 17  2.8
 Annual -6.8 @ 4h00 26 Jan 32.1 @16h00  9 Jul 10.4
 ---Seasons & typical periods---
Winter season is Mon  1 Jan - Sun 11 Mar
Typical winter week begins Mon 19 Feb
Spring season is Mon 12 Mar - Sun 24 Jun
Typical spring week begins Mon 21 May
Summer season is Mon 25 Jun - Sun 26 Aug
Typical summer week begins Mon  6 Aug
Autumn season is Mon 27 Aug - Sun 18 Nov
Typical autumn week begins Mon  1 Oct
Winter season is Mon 19 Nov - Mon 31 Dec
Typical winter week begins Mon 17 Dec
*help_end
```

7 Zone controls

In ESP-r environmental control systems can be represented as either so-called idealised zone/flow controls or as a network of system components which is often called a plant system. The choice of which approach to take is partly based on how much you want to know about the detailed performance of the environmental control system and partly on how much descriptive information you can acquire about the composition of the environmental control.

This chapter focuses on idealised zone controls. ...

This chapter will be completed at a later date.

8 Thermophysical resolution

In ESP-r there are essential elements of the model description which must exist for the simulator to be invoked. These relate to the form and composition of the model, schedules of occupancy, lighting and small power and optionally environmental control systems.

And the tag-line for ESP-r has long been:

functionality follows description

This chapter discuss optional facilities to alter the thermophysical resolution of a model so that, for example, radiation exchange is represented explicitly or convective heat transfer is evaluated by an alternative computational approach.

There are many such choices available within simulation tools. They are typically invoked by including additional descriptive terms in the model and/or additional directives to the numerical engine. They are often treated as optional facilities because there are resource (computing and staff time) implications in invoking them.

The user is confronted with the need to understand *the functionality* of such facilities. This chapter will provide an overview of optional facilities available within ESP-r. And, because this is *The Cookbook*, we are going to balance functionality with *methods* for deciding when additional thermophysical resolution is required, *techniques* to determine the resources needed and then back to *methods* for taking advantage of the additional information.

ESP-r, like other simulation tools is over-functional. It is also a general tool that can be coerced to carry out either something close to magic or end up filling up hard disks for little or no benefit to the simulation team. Tool-driven practitioners push buttons because they exist. Readers of the *Cookbook* might be more inclined to be selective in their button pressing.

This chapter will be completed at a later date.

9 Preparation for simulation

In ESP-r the model description can include a number of directives about the nature of the numerical assessments to be carried out as well as where to store the performance predictions for each of the analysis domains. There are a number of reasons why such information is embedded in the model:

- decisions made in the planning stage can be recorded
- clarify working practices
- support for automation and production tasks
- to allow performance predictions to be re-generated at a later date
- information embedded in a model is safer than scraps of paper

The idea of *simulation directive sets* came about because a highly competent user of ESP-r was asked by a client to re-run a historical project to extract some additional information. The archive of the model was found and the notes about the project were scanned and the simulation was re-run. Then business became complicated and frustrating. The predictions did not match what was in the initial client report.

What had changed? Where did the fault lie? Had the model become corrupted? Had the numerical engine changed in a way that would cause different predictions? Eventually, after days of tests the causal factor was found to be the number of *pre-simulation* days that were used. That particular value had not been recorded in the project notes.

And thus it became clear that the description of the model would be much more robust if it included a description of the specific assumptions and directives used by the simulation engine.

One of the side-effects of recording such information is that it simplifies the task of carrying out production runs and reduces the chances of errors.

It also supports *implementing the plan*. If, at the planning stage, the team decide that specific weeks will be a good test of building performance there is a way to record these preferences so that when the model is ready to test is is*very easy* to implement the test.

Further text to be added here.

Integrated performance views

In addition to directives about the nature of the simulation to be run ESP-r also supports directives about the performance issues which the simulation team is interested. The jargon that the ESP-r community uses to describe directives describing what we want to measure and where we want to measure it is *Integrated Performance Views* (IPV). Essentially, this is a formal

language which supports mulit-criteria assessments. Multi-criteria assessments provide a rich set of information that can help simulation teams identify unintended consequences of design decisions as well as identify opportunities for improvements in a design.

How does this differ from the *meters* implemented in other tools? The what and where is a similar idea. The difference is in how the raw values are used after they have been extracted. Each item included in the IPV has a statistical report and tabular data and a summary produced. Multiple report types are generated because different users recognise patterns in different forms.

And the IPV has similar risks to any prior-specification scheme. If it does not include a range of topics it will fail as technique for enforcing multi-criteria assessments.

This chapter will be completed at a later date.

UNDERSTANDING PERFORMANCE PREDICTIONS

10 Understanding performance predictions

As ESP-r workshops have evolved over the years a pattern has emerged. Participants spend almost as much time in the task of understanding the performance of models they created as they did in creating the model. This same pattern is also found in simulation teams that deliver information that exceeds client expectations. Understanding performance predictions is critical for testing that the semantics of the model are correct.

Essentially, we invest our time and energy in the creation of models in order to get to the point where we can explore the temperatures and flux and flow that were generated by the simulation engine. The better our skills as identifying patterns and looking at the chain of thermophysical dependencies within our virtual world the better we can reach that **AH!** point where we can tell someone else a good story about what we have found.

ESP-r differs from other simulation tools in that it records the thermophysical state of the model at each timestep into one or more random access file stores (depending on the number of analysis domains processed by the simulation engine). Some would call these random access file stores databases. The ESP-r suite include a module *res* which is able to recover the values of the thermophysical state from these databases and present them in graphical and statistical and tabular form.

Other simulation tools tend to write out pre-selected performance data and rely on third party applications to parse and process and display such information. There are advantages and disadvantages in the approaches taken by different simulation suites.

The essential difference is of philosophy. In one case you direct the simulation engine what type of data and the location of the data to record before you run the simulation and in the other case the user directives are essentially delayed until the point of data extraction.

Risk: if the simulation engine only records what you ask it to record unintended consequences or opportunities may not be identified

Risk: performance data needed for clarification may not be available and require altering the save directives and re-running the simulation

3 Benefit: for standard reports and for cases where you already know what you want to measure then user directives and third party tools can be more efficient.

The *res* module

Res is a tool for interactive exploration of a rich data store. It allows users to

create ad-hoc collections of information and view them in different formats or perform statistical operations on the data or filter it in various forms. Such facilities are used by practitioners to react to ad-hoc questions and to explore dependencies within the model.

Because the display of graphs and tables and tabular data is optimised for interactive work it tends not to be at a display resolution which is suitable for presentation reports and the typical approach taken is to export the relevant data in a format that can be processed by a third party application.

This chapter will review the types of information that is held in the data store as well as the various *views* of the data that can be generated. And, of course, the core purpose of the chapter is in methodologies of exploration which have been proven successful in identifying unintended consequences and opportunities.

This chapter will be completed at a later date.

11 Flow

This Chapter focuses on airflow modelling techniques in ESP-r:

- overview of network air flow modelling;
- the art of planning flow networks;
- defining and calibrating flow networks;
- control of flow networks;
- project management.

At the simplest level, air flow within an ESP-r model can be imposed via schedules of infiltration and ventilation. For example a user could stipulate 0.5 air changes of infiltration for all days and all hours or a schedule that changes at each hour for each day type. These schedules may be subjected to control (e.g. increase infiltration to 1.5 air changes if zone temperature goes over 24° C). Scheduled flow is appropriate for engineering approximations, initial design studies and basic operational regimes.

11.1 Limitations of Scheduled Flow

The critical word above is *imposed*. Schedules impose a flow regime which may have no basis in the *physics* of buildings. This is particularly true in the following cases:

- Where there is strong coupling between heat and air flow, e.g. stack and wind driven flow
- Where there are highly dynamic variations in ventilation rate, e.g. natural ventilation
- Where control strategies are important, e.g. opening or closing windows based on temperature and/or wind speed.

If these are descriptive of your design then consider flow networks as a way to better represent the dynamics of flow interactions.

11.2 Fluid Flow Networks

Fluid flow networks offer considerable flexibility in describing a range of designs and the potential to increase the resolution of models to support assessments which are dependant on the movement of mass or flux or power within a model.

Rather than imposing flows, a network describes possible flow paths (e.g. doors, cracks, fans, ducting, pipes, valves), points where boundary conditions apply (e.g. an opening to the outside) and locations where measurements of flow performance are required (e.g. internal and boundary nodes).

ESP-r's flow network solver dynamically calculates the pressure-driven flows within zones and/or environmental control systems that are associated with the network. The flows at nodes are a function of nodal pressures and

the connected components' characteristics. The mass balance at each node is solved using a customised Newton-Raphson approach. The solution iterates until it converges and the pressure at each node and the mass flow through each connection are saved.

The solution takes into account the change in driving forces as conditions within the building and boundary conditions evolve. Information exchange between the domain solvers ensures that changes in flow will influence conditions in associated zones, controls, systems and CFD domains within a model. For example, opening a window on a cool day introduces cool air into a zone, which depresses the temperature of the walls which then alters the buoyancy forces driving the flow.

Although the flow of air in real building continually adapts to conditions and *self-balances*, simulation re-evaluates conditions at fixed intervals. Thus there is the possibility that differences in temperatures can build up during the simulation time step which would not be observed in real buildings. This results in exaggerated air flows, or oscillating flows, especially for large openings between thermal zones. Thus the choice of simulation time step is a topic worthy of discussion.

Those who wish to know more about the solution technique should look at publications page on the ESRU web site. For example, *On the Conflation of Contaminant Behaviour within Whole Building Performance Simuilation* (A Samuel, 2006), *Energy Simulation in Building Design* (J A Clarke 2001), *The Adaptive Copupling of Heat and Air flow Modelling Within Dynamic Whole-Building Simulation* (I Beausoleil-Morrison 2000), *On the thermal interaction of building structure and heating and ventilating system* (J L M Hensen 1991) are books or PhD thesis which discuss some aspects of flow simulation.

An ESP-r model may have one or more de-coupled flow networks - some describing building air flow and others representing flow in a hot water heating system. In models which include separate buildings the flow network can include one or more of the buildings.

The solver is efficient and thus, even with scores of nodes and components, there is only a slight increase in computational time. Assessing thousands of time-steps of flow patterns in support of natural ventilation risk assessments is one use of flow networks. Flow networks are often used as pre-cursors to CFD studies.

The flexibility of flow networks brings both power and risk. The discussion that follows presents methodologies to help you decide when networks are called for, planing tips for flow networks and techniques for understanding the predicted patterns of flow within your model.

ESP-r differs from some other simulation tools in that it asks the user to define flow networks explicitly. An explicit description allows knowledgeable users to control the resolution of the network as well as the choice of flow components and their details.

This approach assumes that the user has opinions about flow paths as well as access to relevant information about

the flow components used in the network. As with the geometric resolution of a model, creating flow networks is as much an art as it is a science.

Scaling up skills and working practices to cope with the level of complexity found in realistic projects has traditionally been accomplished via mentors or workshops. This has limited the deployment of the facility.

The *Cookbook* aims to de-mystify the topic of flow networks as well as providing hints about where the dragons tend to hide.

11.3 Building blocks

The building blocks we can use to define a mass flow network are flow nodes, flow components and flow connections. The least complex network that the solver will work with includes a zone node and two boundary nodes connected by two components (see Figure 11.1).

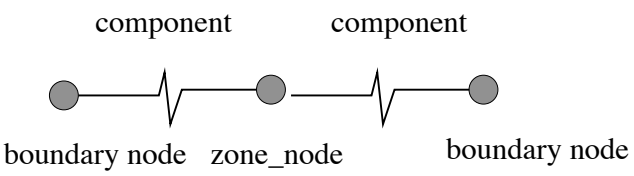


Figure 11.1: The least complex network.

Flow Nodes

Nodes in a flow network are measuring point for pressure, temperature and rate-of-flow. These bookkeeping entities are of four types:

- internal unknown pressure;
- internal known pressure (rarely used);
- boundary known pressure (rarely used);
- boundary with wind induced pressure (air only).

A flow node has one temperature just as the volume of air associated with a thermal zone has one temperature. Typically there will be a one-to-one mapping between thermal zones and flow nodes. Internal nodes can either take their temperature from a thermal zone or they can be defined to track the temperature of a specific flow node. Lets call the former *real nodes* and the latter *extra* nodes.

Wind induced boundary nodes represent wind pressure at one point on the facade of a building. It is a function of the wind velocity, direction, terrain, building height, surface orientation and position within the facade. Typically, a flow network would include a boundary node for each location where air may flow into a building.

Good practice places a boundary node at the height of each opening to the outside. If you follow this pattern the process of locating flow components representing the opening is simplified.

Pressure coefficients

The jargon we use to express changes in pressure due to changes in wind direction is a pressure coefficient Cp. In ESP-r, Cp values for standard angles of incidence (16 values to represent 360°) for a specific location are held as a set.

ESP-r has a database of Cp coefficient sets for different types and orientations of surfaces derived from the literature.

Lets be clear about this, one of the greatest points of uncertainty in flow simulation is in the derivation of pressure coefficient sets. Some groups reduce this uncertainty by undertaking wind tunnel tests or creating virtual wind tunnel tests via the use of CFD. ESP-r offers a so-called Cpcalc function to generate pressure coefficients but it is one of those places where dragons live.

Wind Speed Reduction Factors

The ratio of the climate wind velocity (usually at a standard height of 10m) and wind in the locality of the building is known as the wind speed wind reduction factor such that

```
V = Vclim x Rf
```

Rf is calculated from some assumed wind speed profile and accounts for differences between wind measurement height and surrounding terrain (urban, rural, city centre) and building height and surrounding terrain.

Wind profiles can be calculated using three different models: powerlaw, LBL, logarithmic.

Caution is advised in the use of Rf values as the profiles they are derived from are invalid within the *urban canopy*. In such cases, it is advisable to use a small value for Rf in cooling/air quality studies and high values for infiltration heating studies to represent worst case conditions.

11.3.1 Flow components

Flow components (e.g. fans, pumps, ducts, cracks, valves, orifices etc.) describe the flow characteristics between flow nodes. Flow is usually a non-linear function of the pressure difference across the component based on experimental and analytical studies taken from the literature.

ESP-r has a set of in-built flow components including ducts, pipes, fans and pumps as well as cracks, orifices and doors. Generic fixed volume or mass flow components as well as quadratic and power law resistance models are also included. Details of the methods used are included in the source code in the folder src/esrumfs. Some commonly used components (reference numbers in brackets) are:

- Power law volume flow (10) can be used where flow is well described by a power law
- Self regulating vent (11) is a European vent to embed in a window frame which moderates flow across a range of pressures
- Power law mass flow (15 & 17) can be used where flow follows a power law.
- Quadratic law volume (20) and mass (25) can be use where flow follows a quadratic fit.
- Constant volume flow (30) and mass flow (35) are abstract representations of a fan (can be controlled to approximate variable speeds).
- Common orifice (40) can be used for openings with a user defined discharge coefficient
- Specific air flow opening (110) has a fixed discharge coefficient and only requires an area.
- Specific air flow crack (120) is useful for openings up to 12mm wide.

- Bi-directional flow component (130) is useful for doors and windows which are door shaped where temperature and pressure differences can result in flow in two directions.
- Roof outlet/cowl (211) is based on measurements of typical ceramic units found in Europe.
- Conduit with converging 3-leg (220) and diverging 3-leg (230). There are lots of parameters needed to describe this and dragons live here.
- Conduit with converging 4-leg (240) and diverging 4-leg (250). There are lots of parameters needed to describe this and dragons have been sighted here.
- Compound component (500) points to two components e.g. an opening and a crack with parameters needed for control actions.

11.3.2 Flow connections

Flow connections link the nodes and the components via a descriptive syntax that takes the form: boundary node 'south' is linked to internal node 'office' via component 'door'.

A connection also defines the spacial relationship between the node and the component e.g. that a floor grill is 1.5m below the node. If nodes and connections are at different heights, then the pressure difference across the connection will also include stack effects.

Good practice places a boundary node at the height of each opening to the outside. If you follow this pattern the delta height is zero for components from the point of view of the boundary node.

In the upper part of Figure 11.2 is a crack under a door that connects to zones (where the zone nodes are at different heights). In the lower part of Figure 11.2 is a representation of a sash window (where there is one flow component representing the upper opening and another representing the lower opening. The sash window is associated with two separate boundary nodes.

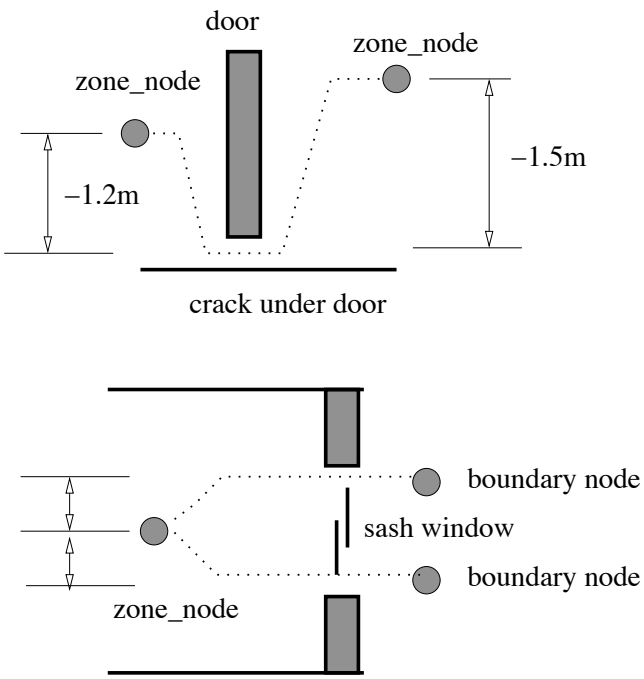


Figure 11.2: A crack under a door and a sash window.

Some people find the syntax used to describe the relationship between nodes and components a bit of a challenge. Here is one technique that works for quite a few users.

> Imagine yourself at the position of the flow node (e.g. in the centre of the zones air volume) and looking at the component. If you are looking horizontally then the height difference is zero. If you are looking up then the height difference is positive. If you are looking

down (e.g. to the crack under the door) then the height difference is negative.

For example, if a zone node is at 1.5m above the ground and the lower window opening is at 1m above the ground and the upper window opening is at 2.1m above the ground. Look at the lower window opening from the point of view of the zone node it is 0.5m below the zone node and at the same height as the lower boundary node. The upper window opening is 0.6m above the zone node and at the same height as the upper boundary node.

Path to boundary

The other rule about creating networks is that every internal node must somehow have a path to a boundary node. The solver treats air as incompressible (for all practical purposes) and the solution is of the **mass** transfer within the network. When the temperature changes the volume must change and if the pressure has no point of relief then the solver breaks.

The design of the network must take this rule into account, especially if control applied to components would have the effect of totally isolating a node. The risk of breaking the rule increases with network complexity so the *Cookbook* emphasises working procedures which are robust enough to scale with the complexity of the models we create.

Parallel and sequential connections

Figure 11.3 is a portion of a network which uses a common orifice to represent the window when it is opened as well as a parallel path with a crack. If the window is controlled and open then the crack has little or no impact on predictions. When the window is closed then the crack becomes the connection to the boundary node.

Recent versions of ESP-r support the concept of a compound component which is made up of, for example, an opening and a crack. The use of compound components can simplify flow networks by reducing the need to parallel connections.

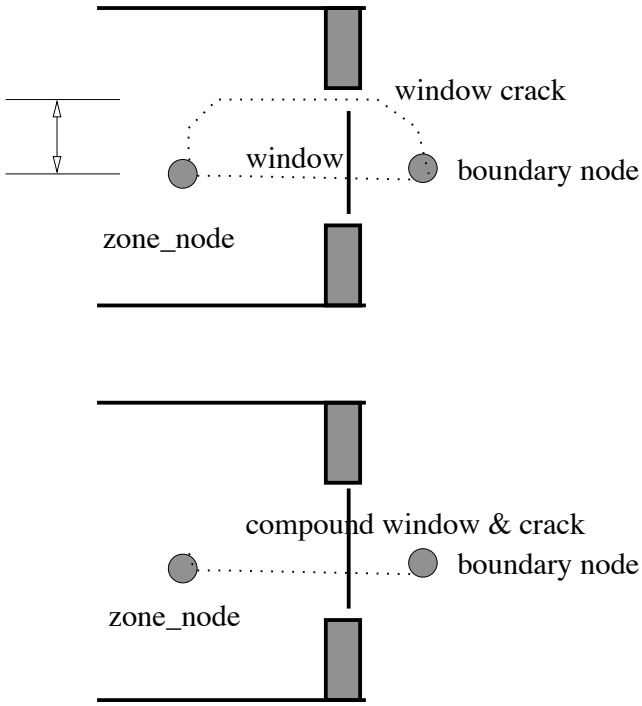


Figure 11.3: A crack & window and a compound component.

Up to this point the networks have been linked via a single component or components in parallel. Suppose we wanted to open a window if the temperature at the zone node was above 22°C AND the outside temperature was below 19°C. How might we implement the above AND logic? One technique is to use components in series.

If *zone_node* is linked to *boundary_node* by component *window* and we wish to insert a second component *control* it would be necessary to create *extra_node*, assign it to track the temperature of *zone_node*. The new flow component *control* should be of a type that would present little resistance to flow when open. The existing connection would be re-defined and the second connection in the sequence would be created as in Figure 11.4. The *window* component would take one control logic and the *control* component would take the second control. The original window crack connection would remain.

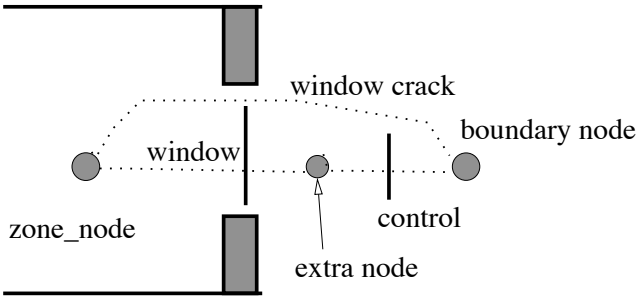


Figure 11.4: Use of additional nodes and components for control

When control is imposed, consideration should be given to adjusting the simulation time step to take into account the response of the sensor and flow actuator. If we observe flow rates oscillating it is usually in indicator that we need to shorten the simulation time step.

11.4 Steps in creating a network

The *ESP-r Cookbook* recommends a methodical approach to the creation of network. It is possible to design networks of dozens of nodes and components which work correctly the first time. Plan carefully and then implement the plan! Successful practitioners use the following set of rules:

Rule one
: sketch out the network either as a 2D layout or as a 3D overly of the wire-frame view of the model

Rule two
: give informative names to the nodes and components on the sketch and use these same names when using the interface

Rule three
: identify portions of the network where control will be applied

Rule four
: if there are likely to be design variants, use overlays on the sketch to lay out alternatives and ensure that the sketch provides a summary of the intent of the overlay

Rule five
: if there is room on the sketch include critical attributes of the components, if there is not ensure you separately record component attributes before you start on the interface

A good sketch is worth hours of debugging - trust us on this!

ESP-r does not graphically represent the network (yet). The sketch makes it easier to interpret the story told by the names of the nodes and components.

When we plan a network we should account for how flow is induced or restricted, where control might be imposed as well as where we expect

there to be differences in air temperature within a building.

For example, if you *believe* that a perimeter section of an office will often be at a different temperature consider creating a separate zone for the perimeter. Some practitioners include additional vertices in the model and subdivide floor and ceiling surfaces in their initial model so that it is easy to subdivide a zone at a later stage.

Experience indicates that last minute subdivision of zones can take longer than one might expect and require time for testing that has not been budgeted for.

The design of the flow network should take into account what-we-want-to-measure. Typically there is a one-to-one matching of internal nodes and thermal zones. However, if there were many openings in a room on the east and to the south facades and we were interested in the aggregate flow at each facade then we might plan a network which had extra internal nodes as shown in Figure 11.5. In the sketch the *west node* and *east node* nodes take their temperatures from *zone_node*.

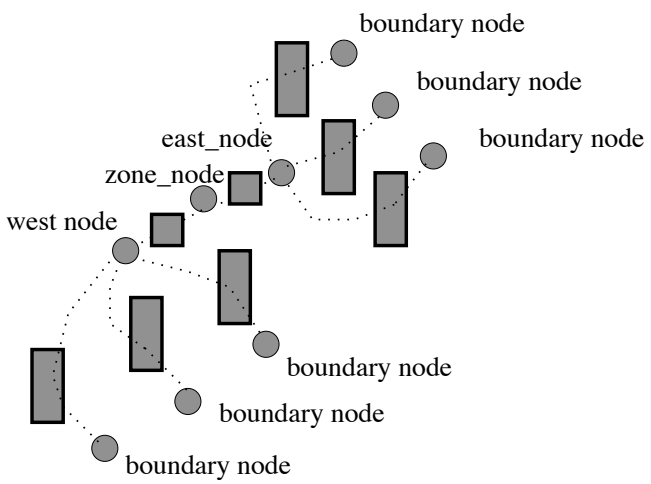


Figure 11.5: network with extra nodes

Linking the nodes and components is a critical step which benefits from a methodical approach as well as consistent pattern for typical relationships.

> A good sketch is still worth hours of debugging!

In deciding which components to use our initial task is to review the features of the available components and control options which can be applied to them as well as their data requirements.

There is also the option of browsing existing models which include flow networks and running assessments and reviewing the predictions. A great way to understand how flow works is to take these exemplar models and systematically adapt the network and/or control description and observe the changed performance.

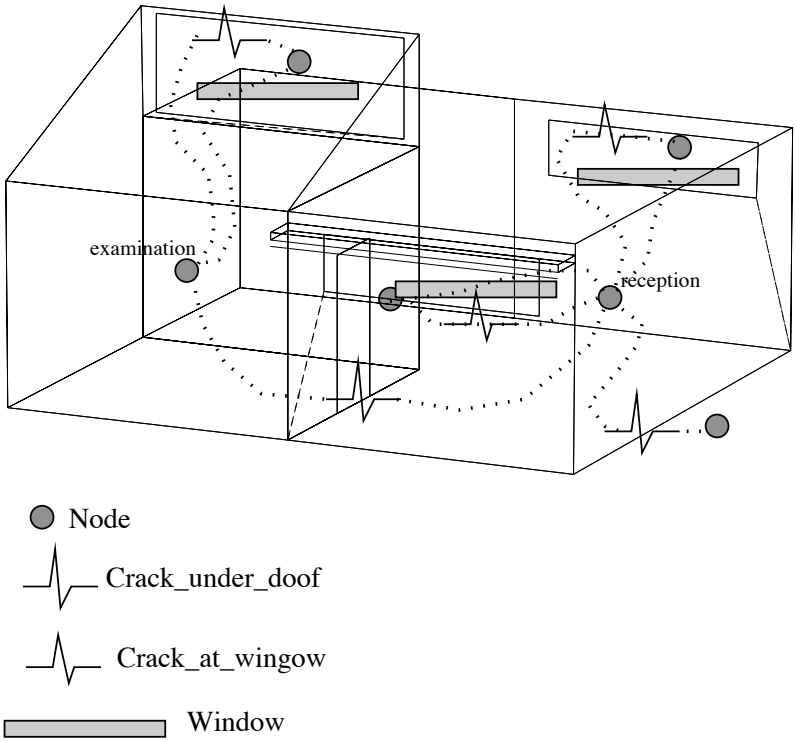


Figure 11.6: simple network for doctor's office

11.5 A simple network

To illustrate the process lets create a simple network for the doctors office that you worked on earlier.

Where could air flow

Air can flow under the door between the reception and the examination rooms. When we created the initial zone a decision was made to exclude the door as being unimportant in terms of the overall heat flow in that portion of the building. An entrance to the reception would be place where air would flow. Does this require that we go back into the geometry and add a door? Not necessarily.

Consider if there was a 5mm crack in the wall. The handful of missing mortar would not change the thermophysical state of the wall, but the air moving through the crack might be noticed. In ESP-r the description of the zone and the flow network can *differ* as long as we remember the *intent* of the difference.

Air can also flow around the frames of the windows. It is difficult to know if there are differences in the leakage paths of the three horizontal windows. For this project lets assume that they have the same leakage characteristics. There are three locations on the facade with these windows - but each can be represented by the same crack component. The north window is a different size and lets assume that it has a different leakage characteristic so that needs to be a different component.

The next issue is how long is the typical crack and where is it. If we wished to be pedantic there are cracks at the top and bottom and left and right of each window. For this exercise lets

assume the crack is along the centre line of the window and is the same length as the width of the window.

What kind of window opening?

If someone was to open a window then there might be flow IF there was someplace for the air to go. A mass flow network has limitations when attempting to represent single sided ventilation. In some cases it is possible to use a bi-directional component to represent a window (there are arguments in the community about the validity of such an approach). If two windows are opened the driving forces are better represented within a mass flow network.

Depending on the size and physical details of the opening there are several flow components which might be useful. And before we look at the component details it is worth considering whether the windows in this doctors office will be operated by the staff and whether they will do this with a predictable logic (e.g. will they open the window if it reaches 22°C and will they open the window fully)?

For this project lets start with the assumption that we will eventually want to test out a window opening regime that involves opening the windows twenty five percent. We can include the relevant components in our planning so that they will be easy to include at a later date. How the window operates will determine the types of flow components we should use as well as the connections we establish within the flow network.

ESP-r offers a pre-defined window component which only requires the opening area, it also includes a common orifice which requires an area and a discharge factor. If one had the data the opening could be a polynomial or a power law. A sash window is two openings and some users also represent a horizontally pivoting window as two separate openings. Some windows have an width to height which is similar to a door and in that case some users may decide to use a bi-directional flow component (more on that later). For now, lets use the air flow opening component and assume that the component is centred within the window surface.

What kind of extract fan?

Air can be induced to flow by an extract fan. And here our choices begin to expand because there are both abstract and explicit components for the extractor. And since there are times when the extractor is not required we will need to control it.

Early in the design process our tactic should be to confirm whether forced ventilation in the doctors office will help and the magnitude of the flow required and perhaps the turn-on set point. This is most quickly accomplished via an abstract volume flow rate component and a simple control law. Once we confirm the general characteristics and response of the building we can consider whether a specific fan curve should be applied to the model.

What kind of room?

What else might require our attention? Looking back at the dimensions of the

doctors office the sloped ceiling of the examination room increases the volume of the space and raises the centre of the air volume in comparison to the examination room. The earlier decision to represent the examination room as a single volume of air which is well-mixed (i.e. at a single temperature) does introduce an element of risk because there will be times when the air near the peak of the ceiling is at a different temperature to the occupied portion of the space.

Given the initial design brief the single volume decision was appropriate. If the question becomes occupant comfort and fine-tuning of window opening or extract fan use then a subdivision of the physical space into multiple thermal zones could be done. For the current project lets not alter the zones.

When we sketch out the network we will need names for the nodes and the components (Figure 11.6). At some point we must record the attributes of the components so that this information will be available when we create the network and we will also want to pass our notes to the person checking the model.

This is also a good time to get out a calculator and focus the project manager on the geometry of the zones so that we can record the height differences between the nodes and components. The table below provides several of the relevant dimensions:

Component names and locations.

node name - type height
reception - internal 1.5m
examination - internal 2.25m
south - boundary 2.375m
north - boundary 2.375m
exam_north - boundary 3.75m
exam_extract - boundary 3.0m
east - boundary 0.1m

component name - type data
long_win - air flow opening 1.0m^2
long_cr - crack 5mm x 3.0m
door_cr - crack 10mm x 0.8m
upper_win - orifice 1.5m^2 0.5 coef
upper_cr - crack 5mm x 3.0m
extract - volume flow ~1 air change

node - to - node component
south to reception via long_win
south to reception via long_cr
north to reception via long_win
north to reception via long_cr
east to reception via door_cr
exam_north to examination via upper_win
exam_north to examination via upper_cr
reception to examination via door_cr
examination to exam_extract via extract

height differences:
long_win is 0.875m above reception
long_cr is 0.875m above reception
upper_win is 1.5m above examination
upper_cr is 1.5m above examination
door_cr is 1.5m below reception
door_cr is 2.25m below examination

Which connection comes first?

In addition to what we include in the sketch we also need to consider the order that we link the nodes and components. The connection associated with the extract fan should being at a zone node and end at a boundary node so that the flow is outwards. For other connections the order is not

numerically important.

It does help to have a consistent policy with openings to the outside because a flow along a connection is reported as a positive number if it is in the same direction as the connection definition and negative if it is flowing from the end point to the start point. If you conceptually consider air entering the building as positive flow then you would choose to define connections with the outside as beginning at the boundary node and ending at the zone node.

Our task is to record our decisions and attributes on the flow network sketch and then to use the interface to first define the nodes and then the components and then to link the network together.

11.6 To the keyboard

Having planned and recorded the information associated with our network we can now use the interface to increase the resolution of the model. In the *Project manager* look for `Model management -> browse/edit -> network flow` and choose `flow network (menu)` and confirm the suggested file name and then choose `make new file` and then specify that the network is `all air.` And you will be presented with the initial menu as shown in Figure 11.7. At the start there are no nodes or components or connections and no nodes have been linked to the zones. The descriptive sequence is to define the nodes first and then define the components and lastly the connections between the nodes.

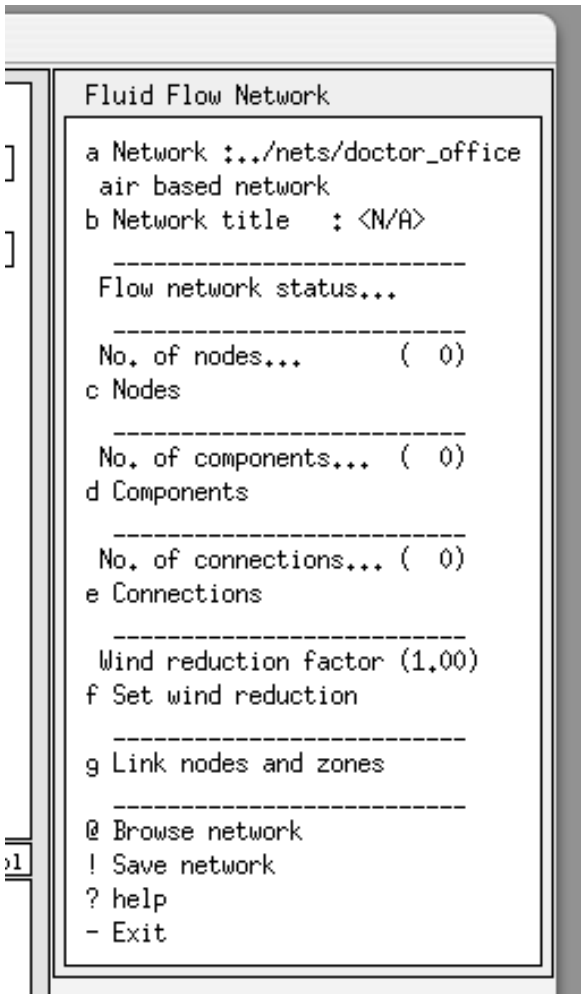


Figure 11.7: interface at start of process

Initial nodes

Select the `nodes` option and allow an auto-generation of the nodes for the current zones. The initial list includes the two items shown in Figure 11.8. The auto-generate assumes that each thermal zone will have a node and the zone name is given to the node and the centre of the zone becomes the height of the flow node.

```
@strath.ac.uk
Nodes
  Name          |Fluid |    Type     |Height |  Data1 |   Data2
a reception      air     internal       1.50      0.0     120.0
b examination    air     internal       2.25      0.0      60.0
  ------------------------
+ add/delete/copy node
? Help
- Exit
```

Figure 11.8: auto-generated nodes

```
@strath.ac.uk
Nodes
  Name          |Fluid |    Type     |Height |  Data1 |   Data2
a reception      air     internal       1.50      0.0     120.0
a reception      air     internal       1.50      0.0     120.0
b examination    air     internal       2.25      0.0      60.0
c south          air     bound wind P   2.38      9.0     180.0
d north          air     bound wind P   2.38      9.0       0.0
e exam_north     air     bound wind P   3.75      9.0       0.0
f exam_extract   air     bound wind P   3.50      9.0      90.0
g east           air     bound wind P   0.10      9.0      90.0
  ------------------------
+ add/delete/copy node
? Help
- Exit
```

Figure 11.9: adding boundary nodes

```
uiries to esru@strath.ac.uk
Components
  Name        |Type| Description ...
a long_win     110  Specific air flow opening          m = rho.f(A,dP)
b long_cr      120  Specific air flow crack            m = rho.f(W,L,dP)
c door_cr      120  Specific air flow crack            m = rho.f(W,L,dP)
d upper_win     40  Common orifice flow component m = rho.f(Cd,A,rho,dP)
e upper_cr     120  Specific air flow crack            m = rho.f(W,L,dP)
f extract       30  Constant vol. flow rate component   m = rho.a
  ------------------------
+ add/delete/copy component
? Help
- Exit
```

Figure 11.10: the completed components

Boundary nodes

The next step is to define the boundary nodes. Do these based on the information in the table above. For each boundary node you will be asked to nominate a surface where the opening is found. This sets the orientation of the boundary node so that wind directions can be resolved. The centre of the surface becomes the height of the node. When asked to confirm the height check the notes (aren't you glad you made a note of this). You will also be asked about which pressure coefficient set to use and for this exercise select `1:1 sheltered wall` for each connection. The result should look something like Figure 11.9. This is a good time to save the network!

Components

The next task is to create the components. Do this based on the information in your notes. Each component requires a name and, depending on the type of component, there will be one or more attributes to define. The order you create the components is not important. When you define the extract fan as a constant volume flow rate component note the various ways you can define the flow rate. The air changes per hour is particularly useful at an early stage so use that (which is equivalent to 0.01667m^3/s based on the volume of the examination room).

When the components are completed they should look similar to that shown in Figure 11.10. Save your network again. You may also want to use the `browse network` option to review the data that you have provided.

Connecting nodes and components

The next step is to link the nodes and components together to form the network. Each connection has an initial node and component and a second node. You are asked to specify the height difference to the component from the point of view of each node. The order you give is not critical except for the extract fan. Flow in the direction of the initial to the second node is reported as a positive number.

It does help to have a consistent pattern when defining connections. One common pattern for connections which involve boundary nodes - use the boundary node as the initial node so that flow into the room is reported as a positive number. Some users define all of the links with boundary nodes first and then they define links between internal nodes.

Another pattern is in the definition of height differences. The boundary nodes were defined at the height of the opening so that the delta height from the position of the boundary node is always zero. And if a connection uses a component where buoyancy will not be an issue (like the constant volume flow component) give zero as the height difference.

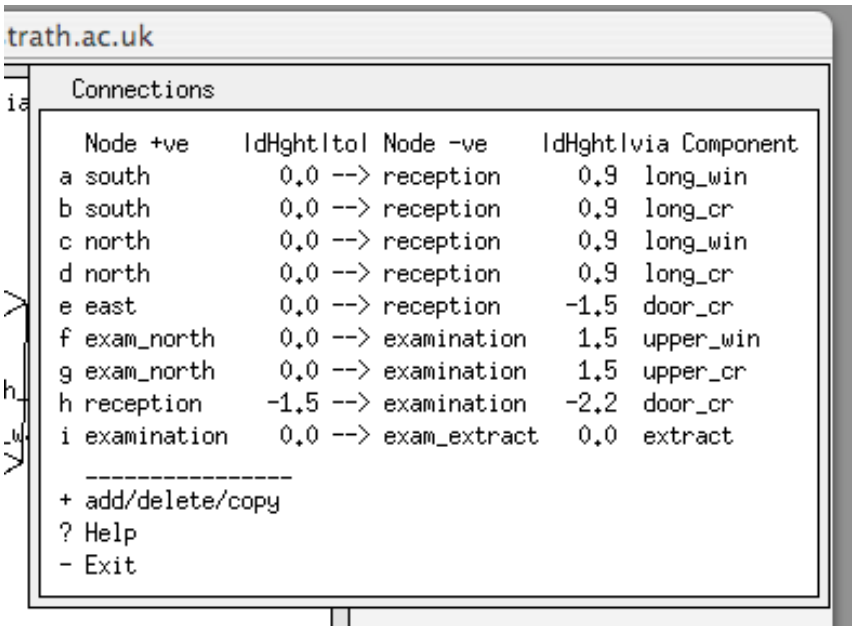


Figure 11.11: the completed connections

```
   7    6    9    1.000    (nodes, components, connections, wind reduction)
Node          Fld. Type   Height    Temperature    Data_1       Data_2
reception       1    0   1.5000       20.000       0.0000       120.00
examination     1    0   2.2500       20.000       0.0000       60.001
south           1    3   2.3750       0.0000       9.0000       180.00
north           1    3   2.3750       0.0000       9.0000       0.0000
exam_north      1    3   3.7500       0.0000       9.0000       0.0000
exam_extract    1    3   3.5000       0.0000       9.0000       90.000
east            1    3  0.10000       0.0000       9.0000       90.000
Component    Type C+ L+ Description
long_win      110  2  0 Specific air flow opening           m = rho.f(A,dP)
 1.  1.
long_cr       120  3  0 Specific air flow crack             m = rho.f(W,L,dP)
 1.  0.00499999989  3.
door_cr       120  3  0 Specific air flow crack             m = rho.f(W,L,dP)
 1.  0.00999999978  0.800000012
upper_win      40  3  0 Common orifice flow component m = rho.f(Cd,A,rho,dP)
 1.  1.5  0.5
upper_cr      120  3  0 Specific air flow crack             m = rho.f(W,L,dP)
 1.  0.00499999989  3.
extract        30  2  0 Constant vol. flow rate component   m = rho.a
 1.  0.0166670009
+Node         dHght   -Node         dHght   via Component
south         0.000   reception     0.875   long_win
south         0.000   reception     0.875   long_cr
north         0.000   reception     0.875   long_win
north         0.000   reception     0.875   long_cr
east          0.000   reception    -1.500   door_cr
exam_north    0.000   examination   1.500   upper_win
exam_north    0.000   examination   1.500   upper_cr
reception    -1.500   examination  -2.250   door_cr
examination   0.000   exam_extract  0.000   extract
```

Figure 11.12: the ESP-r flow network file

Take your time to avoid having to re-define connections. For this exercise there are three sets of parallel connections involving the windows and the cracks. The interface will ask for confirmation when you define the

parallel connection. The interface will ask you if you want to auto-generate the connections. For this exercise say no.

> Hint: as you create the connections, mark your sketch so that your progress is recorded. It is easy to duplicate a connection, or even worse, to miss out a connection.

Checking your data

The last step is to confirm the linkages between the flow nodes and the thermal zones. Look for the `link nodes and zones` option in the interface. When you have completed this it should look similar to that shown in Figure 11.11. Save the network again. After exiting the network menu it is also a good idea to generate a new QA report. When looking at the QA report pay particular attention to the Z values for the components. If the two values differ you might have made a mistake about the delta height from each of the nodes.

The description of the network is written to an ESP-r file in the *nets* folder of your model. The file shown in Figure 11.12 is for the doctor's office.

11.7 Calibrating flow models

Having added a flow network to the model, lets see if the model will run and then if the predicted air flows make sense. What might we look for? The windows are open so we would expect significant flows, especially on days with some wind where up to one air change per minute might occur. We want to include in our assessment days that have a variety of wind speeds and directions.

We might also wish to identify some days where overheating is likely so that after we add controls to the windows and extract fan we can simulate the same period when we are looking at how window opening or the use of an extract fan might improve conditions.

For the initial assessment, select a simulation period of a week in April and reset the simulation time step to 10 minutes.

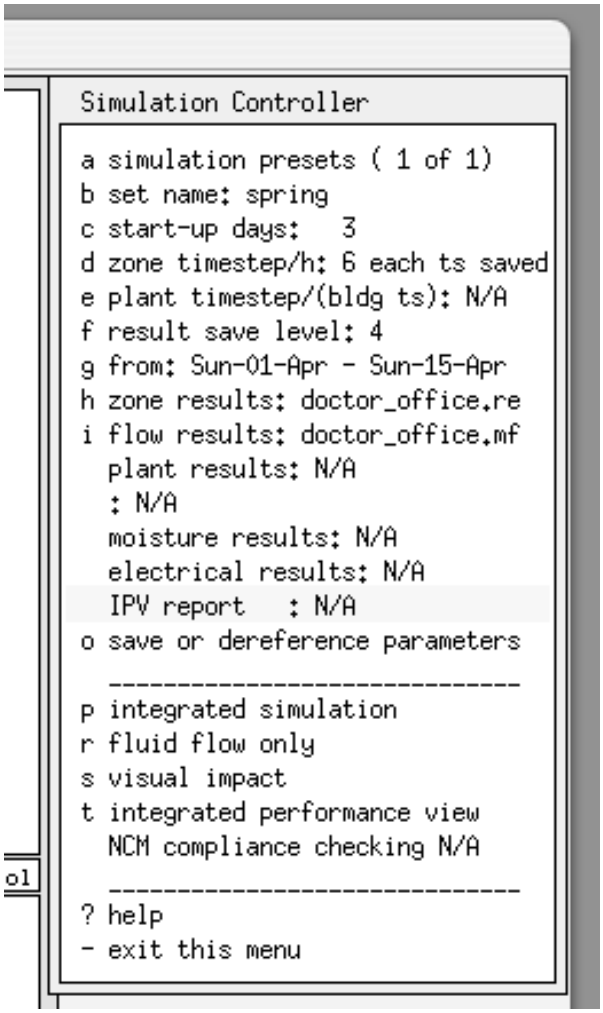


Figure 11.13: simulation parameters for a spring assessment

We we ask for an `integrated simulation` a number of checks will be carried out on the model to ensure that it it both syntactically correct and that the model dependencies are correct.

During the simulation, if the monitoring is turned on (as in Figure 11.14) the inside temperatures is close to ambient (black line) in the reception (where there is cross ventilation) with higher temperatures in the examination room where the flow is constrained by the single sided ventilation and the crack under the door. Clearly there is a need to control the opening of windows to prevent chilling of the room.

Graphs and tables

Before adding control to the flow network, have a look at the predictions of flow to see which of the reports or graphs provide useful performance information. Start of the results analysis module and the last simulation predictions will be used.

A good place to start is the `graph facilities` and the `Time:var graph` where we can look at the energy implications of the air flow. Selecting `climate -> ambient temperature` and `temperatures -> zone temperature` and `zone flux -> infiltration` will provide an overview (Figure 11.15). As expected the infiltration cooling in the examination room is minimal. The infiltration cooling (the energy implication of the air movement) in the reception varies between 0W and 1000W. As the temperature differences decrease the energy implication is reduced.

And it is also the case that the volume of flow changes the magnitude of the energy implication. To look at that we need to use a different graphing facility. Use the `Network flow ->` option and request `volume flow rates -> total entering node` for the two rooms. The flow rates shown in Figure 11.16 should be similar to the pattern you see in your model.

Consider that time spent in the results analysis module working out for yourself which reports and graphs help you to understand the patterns of flow as an investment. Many a practitioner has be caught out by a cursory inspection of flow data!

It is now time to introduce some control into our flow network so that windows are only open when appropriate. And if we find that flows are insufficient then some control of the exhaust fan will be added.

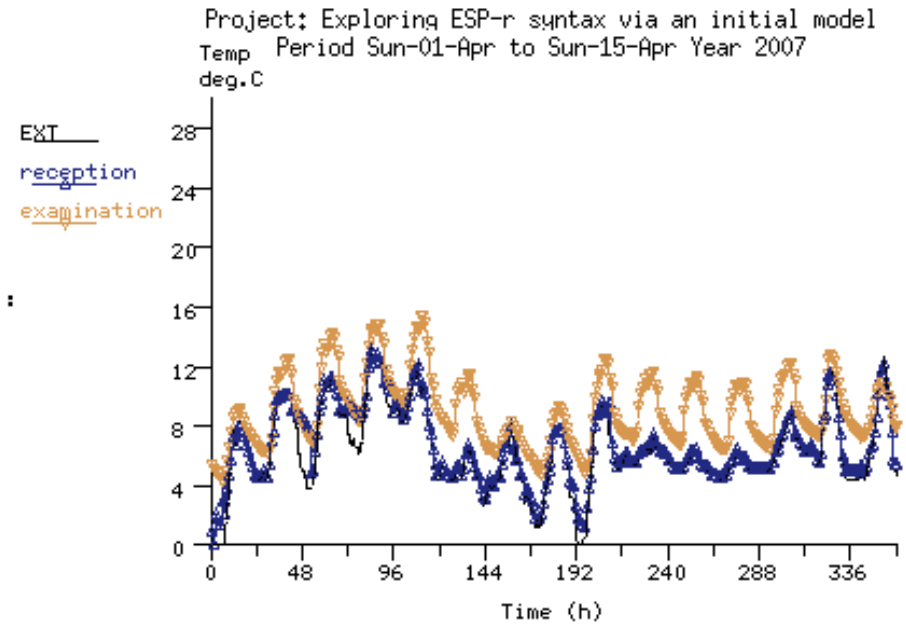


Figure 6.14: temperatures during spring assessment

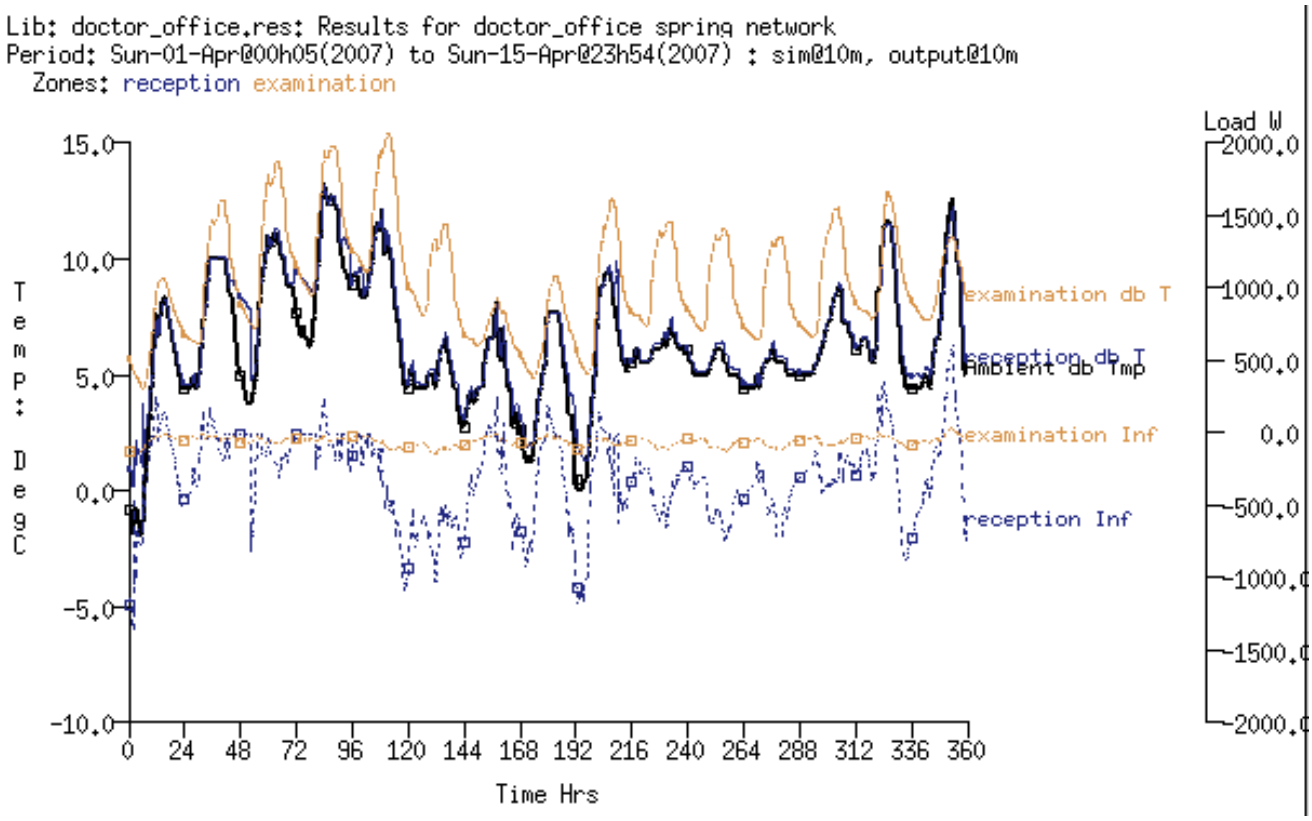


Figure 6.15: temperatures and air cooling during spring assessment

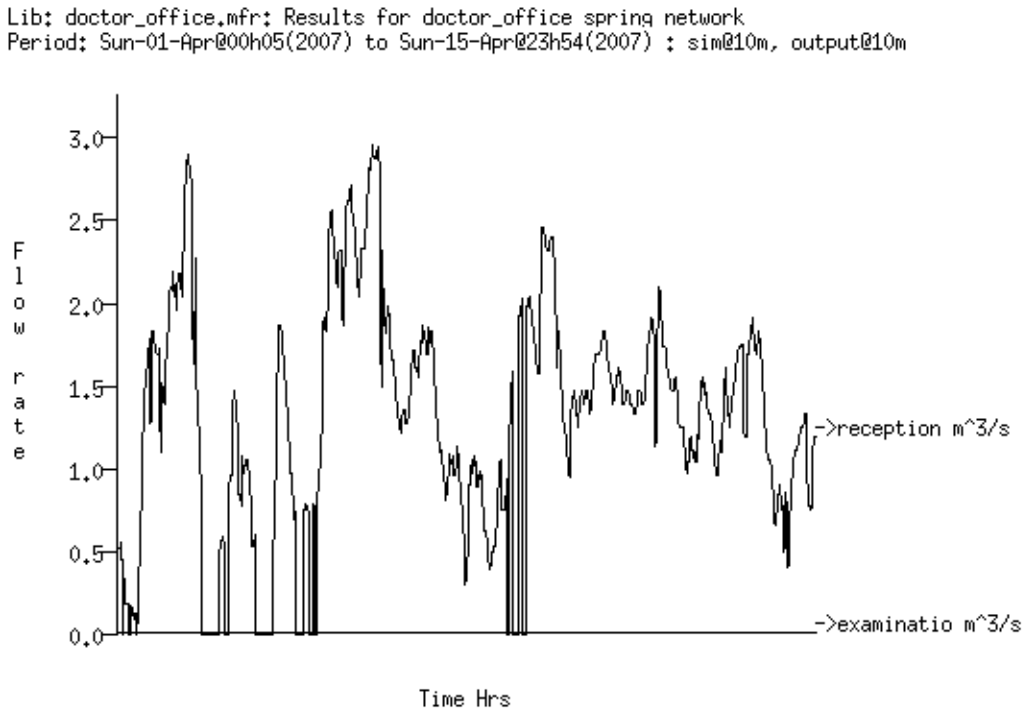


Figure 6.16: temperatures and air cooling during spring assessment

11.8 Flow Control

As with ideal zone control, flow network control uses a *sensor > controller > actuator* structure to define flow control loops. A control loop senses a specific type of data at a specific location, the sensed value is evaluated by a control law and the control actuation is applied to the component associated with a specific flow network connection. A day can be sub-divided into several control periods with different laws or control law details. A number of control loops can be used in sequence or in parallel to implement complex control regimes.

At each simulation time step the flow solver takes the current conditions and predicts the flow. The flow predictions are passed to the zone solver which then generates a new set of conditions to be used by the flow solver at the next time step.

Control logic is tested at each time step. If we are approximating fast acting flow actuation devices then the simulation frequency should reflect this as far as is possible within the constraints of the project. Currently a one minute time step is the highest frequency that is supported by ESP-r for the zone and flow domains.

In terms of the doctors office, a 10 minute time step was used in the uncontrolled version of the model. This will result in a slightly sticky control so we should pay close attention to the predictions to see if this is an issue.

Flow sensors and actuators

In ESP-r a flow sensor is defined by its location and the values which it can sense. These include temperature, temperature difference, pressure, pressure difference, flow rates, humidity, etc. at nodes within the network. It is also

possible to sense temperatures in zones or climate data.

Most flow components are able to be controlled. Control is expressed as a modification to an attribute of the component. For example, a control actuation value of 0.6 applied to a window with an area of 1.0m^2 would result in an opening area of 0.6m^2.

Control is imposed on specific instances of a component so it is possible to control the south facing window in the reception using different control logic from that used for the north facing window.

Flow control laws

The range of control laws includes on/off, range based and proportional control as well as a multi-sensor control where the control law can include AND or OR logic from several sensed conditions.

An ON/OFF control has a single set point and attributes that determine if the control is direct-acting (ON above set point) or indirect (ON below set point) as well as the fraction of the nominal area to use when ON.

A multi-sensor flow control is an ON/OFF controller which includes the definition of more than one sensed location as well as and AND or OR logic to apply. In some cases a multi-sensor flow control can replace a series of individual controls (as described in Figure 11.4).

Range-based control uses the nominal area or flow rate of the component - but switches to an alternative rate/area as a function of the sensed condition - low rate if below low set point; mid rate if above a mid-range set point; high rate if above maximum set point as is shown in Figure 11.17.

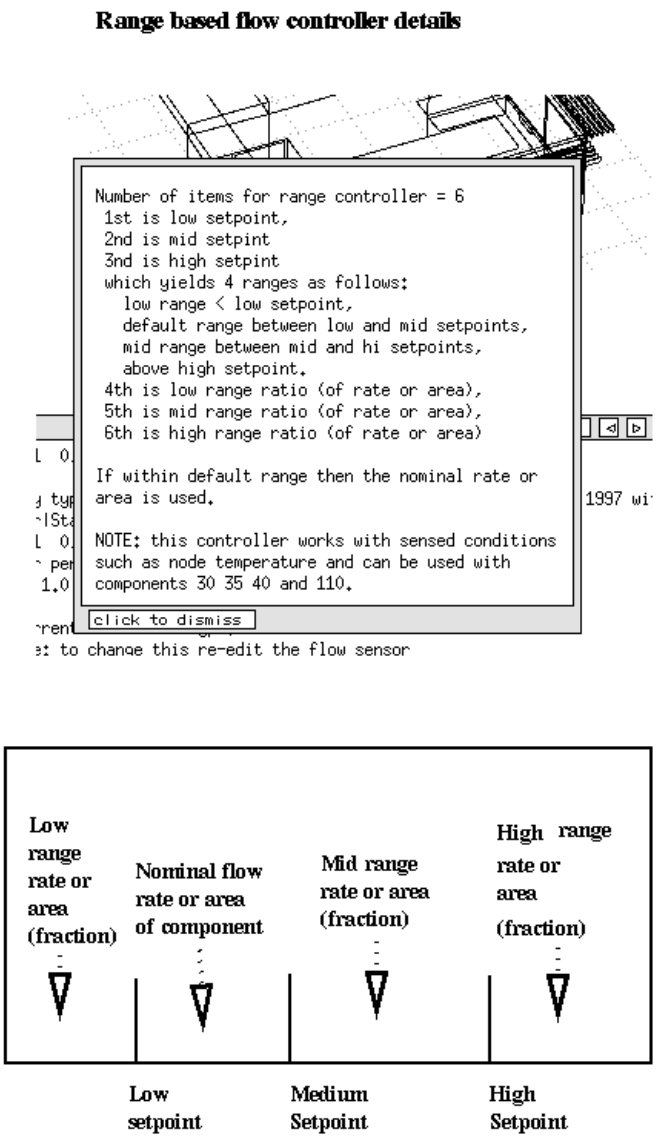


Figure 11.17: overview of range based control

In terms of the doctors office, the initial flow network included window flow components that represented a full-open state. In reality, occupants would probably open a window only as much as was necessary. Unlike automatic controls, occupants would tend to implement a sticky control (i.e. they delay adjustments). What types of control could approximate this?

- ON/OFF with ON expressed as a fraction of nominal opening area - this would be equivalent to closing the window below a set point value and open to X percent of nominal area above this value.
- Proportional control where the percentage opening area varies between two values. Some occupants would not exercise such fine (and continuous) control.
- Range based control could be used for a control that reduced the window opening area if the temperature in the room is close to the heating or cooling set points. In the dead-band between heating and cooling it would allow natural ventilation cooling by first using the nominal area of the window and then additional opening area if required. It would be necessary to re-define the nominal area of the window to represent slightly open rather than fully open.

If occupants are manipulating the windows then the control periods should match the occupied period with an alternative control law for the unoccupied period. Some building may operate a policy of limiting window opening after hours to limit the potential for rain damage. If automatic dampers are manipulating the openings then it would be necessary to find a combination of control law and simulation time step that reflected the regime.

Hint: once you specify a flow control loop it can be copied and then associated with other flow connections.

For our first implementation of flow control lets use an ON/OFF control for each of the windows allowing them to open 25% of their defined area if the temperature rises over 22°C. We will define the control once and then copy the control loop and associate it with the other windows. And lets turn on the extract fan if the temperature in the Examination room goes over 24°C. At night we will keep the windows closed by setting the set point for the windows to 100°C.

11.9 To the keyboard...

First things first. Make a backup of your model. To define flow controls use the `Browse/Edit -> Controls network flow` and accept the suggested control file name. Begin by editing the description line for the flow control and give a synopsis of the flow controls included. Add the first loop which will be used to define the window opening regime (closed at night and weekends and open 25% if over 22°C) during 8h00-18h00 weekdays (Figure 11.18).

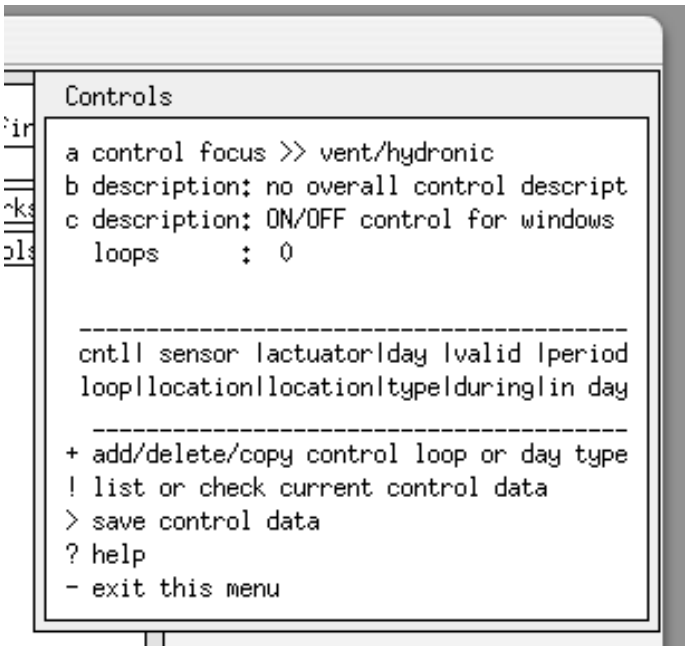


Figure 11.18: control loop interface

prior to adding periods

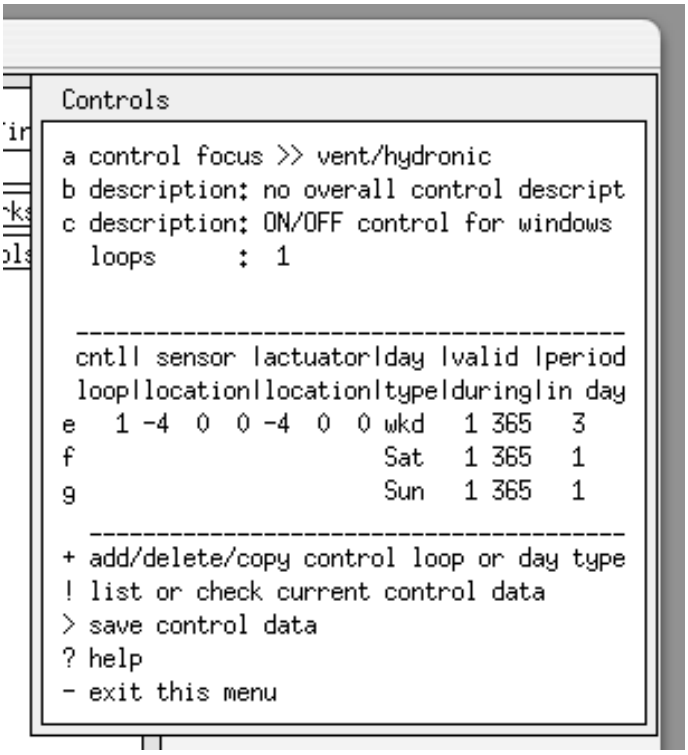


Figure 11.19: control loop interface after adding periods

Select the 'wkd' day type and define the sensor and actuator for the flow connection for the window on the south of the reception (Figure 11.19). The sensor should be located at the reception flow node. And when asked about the flow actuator pick `single flow connection` and then the relevant connection that uses the long_win flow component.

Next edit the period data for the 'wkd'. The first period is an unoccupied period so use a high temperature for the set point. After editing the interface should look like Figure 11.20.

When filling in the data for Saturdays and Sundays there is no need to re-define the sensor and actuator (flow control loops use the same sensor and actuator location for all day types and periods). Once all the day types have been defined save the control file. To use similar logic for the north window in reception copy the first loop and re-assign the sensor and actuator. To use this logic again for the Examination room copy the second control loop and re-assign its sensor and actuator. It is a good idea to keep a note of which control loop is for each of the windows.

The extract fan is probably a simple design with a simple thermostat that does not know what day it is. The control should reflect this by having one day type and one period. This control will sense the temperature at the Examination node and the connection related to the extract fan will be where the control acts.

Update the control file and generate a new QA report and look at the details and your notes to see if the data is correct.

Re-run the simulation. If you turn on monitoring during the simulation you might see something like the Figure 11.21. Now the temperatures in both rooms are much closer and they are both warmer than ambient. If we use the results analysis tool and look at the flow entering the Reception the flow rate is based on the crack component because the temperature did not go above the control set point.

```
ger: enquiries to esru@strath.ac.uk
Control periods
 function  1 day type  1
 number of periods:  3
 ----------------------------------------
 per| start|sensed  |actuated | control law     | data
 no.| time |property|property |                 |
a  1   0.00 dry bulb > flow    on / off          100.00 1.00 0.25
b  2   8.00 dry bulb > flow    on / off          22.00 1.00 0.25
c  3  18.00 dry bulb > flow    on / off          100.00 1.00 0.25
  -------------------------------
* add/ delete a period
? help
- exit
```

Figure 11.20: control loop period details

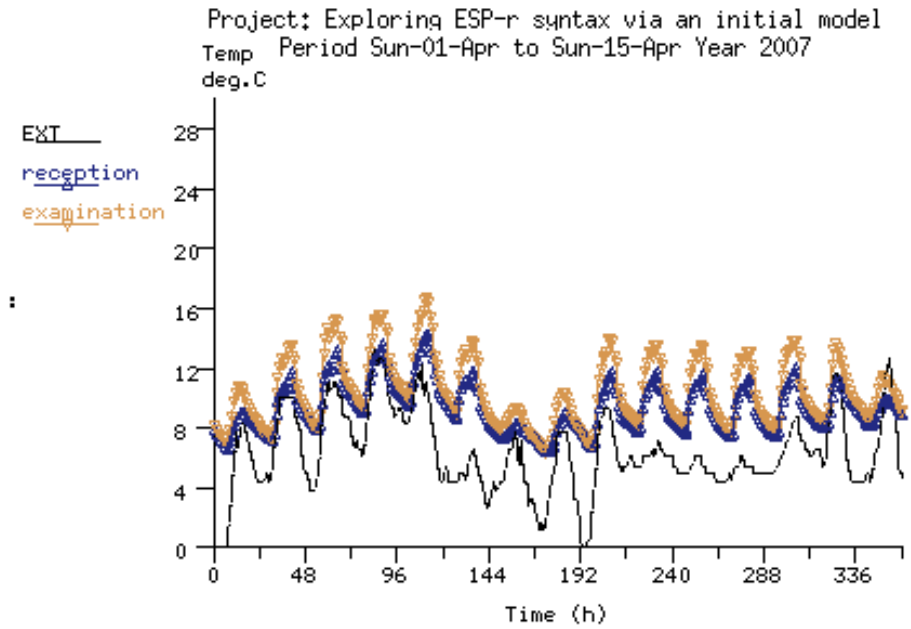


Figure 11.21: spring simulation progress with controlled windows

Select a warmer period of the year e.g. the first week in July and confirm the control is working. When monitoring the simulation (Figure 11.22) we can see that Examination seems to peak at 24°C (which happens to be the set point for the extract fan) and there is some rapid changes in the reception near the 22°C temperature point. Because Examination is constrained by flow under the door it has almost no air movement until the extract fan turns on.

The energy implications of the window opening is clearly seen if we plot `zone flux -> infiltration` in the results analysis module. The windows open briefly except for the last day where they are open for several hours. The extract fan is also on for brief periods except for the warmest days where it also runs for several hours.

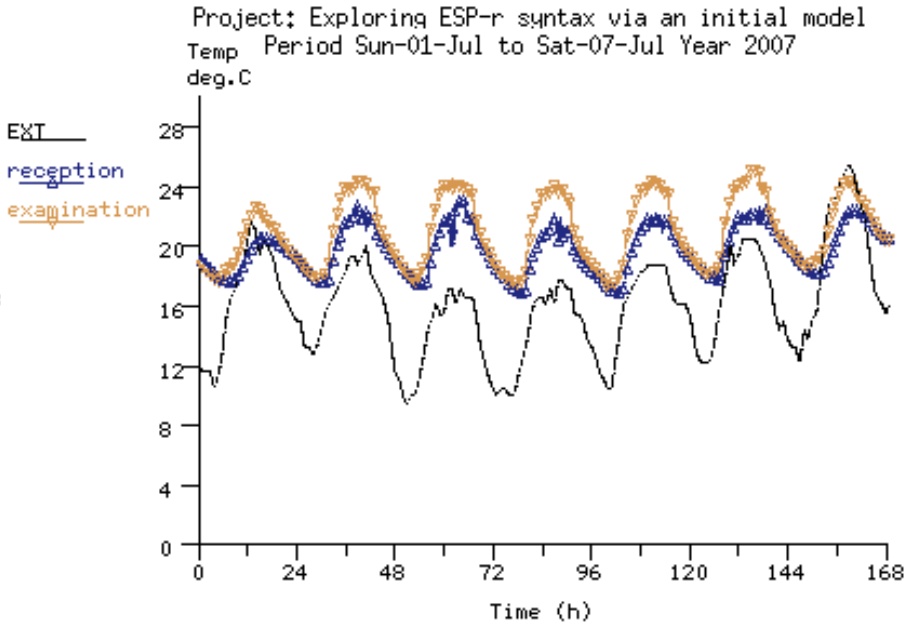
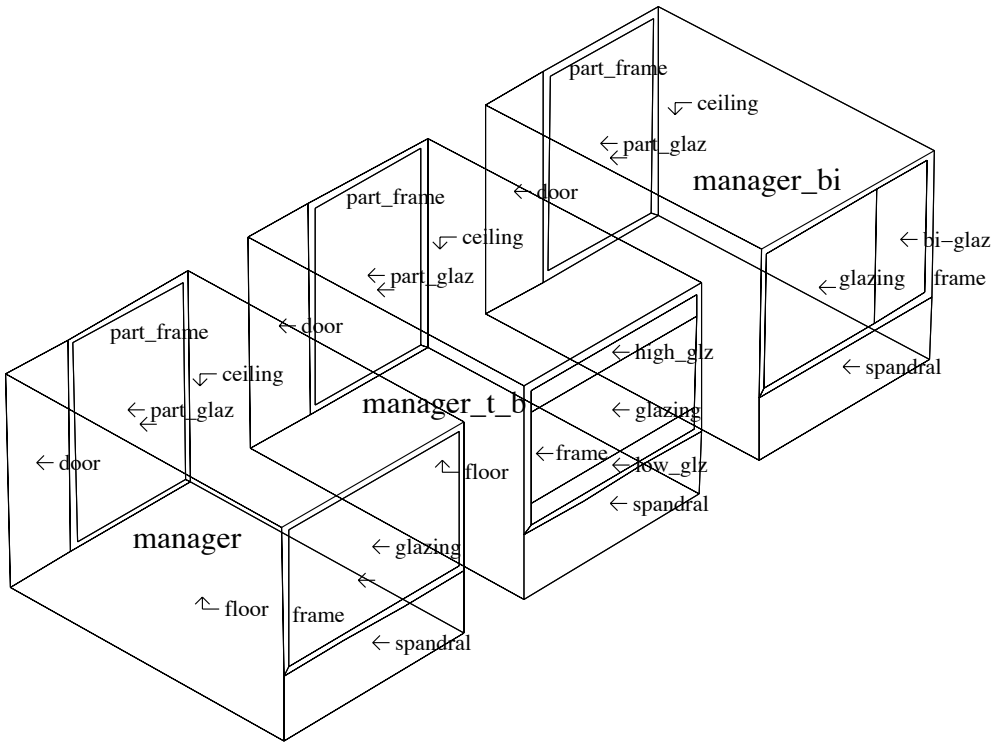


Figure 11.22: warm period with controlled windows



Project: Three offices with different window representations.

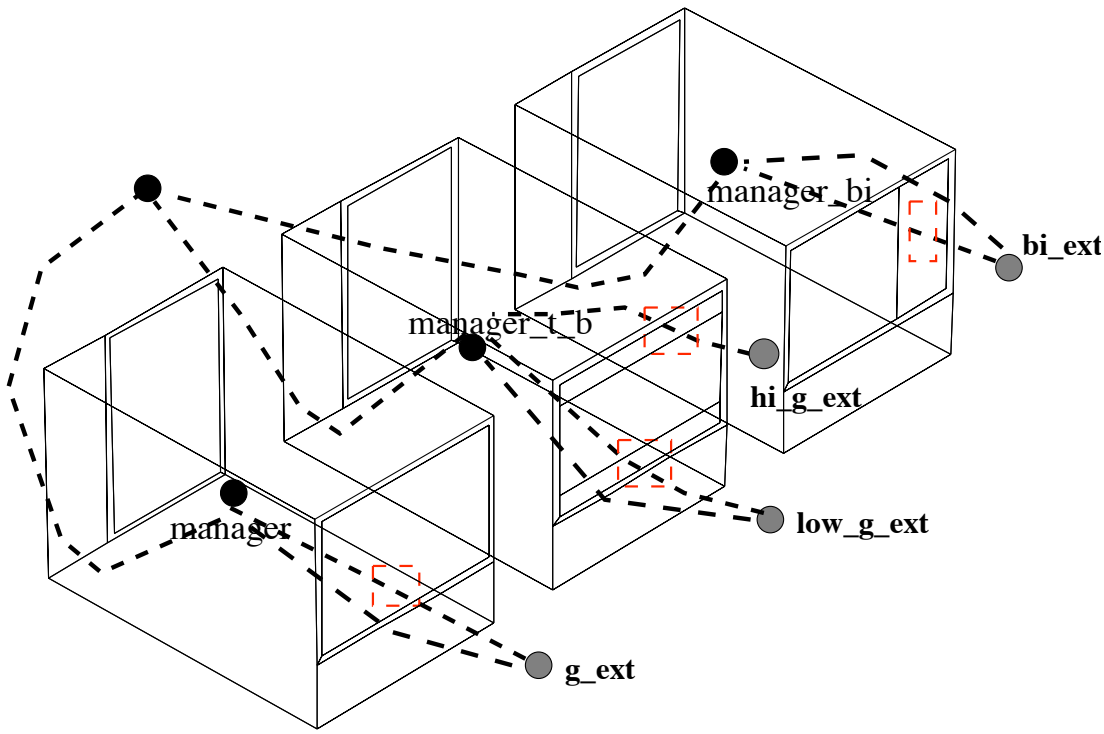
Figure 11.23: wire-frame of three test rooms

11.10 Window representations

This section of the *Cookbook* is work-in-progress.

The first exercise used a simple representation of window openings. Many window types require that we adapt the flow paths and components. The following discussion uses a model with three adjacent zones (Figures 11.23 and 11.24) which differ only in the window details. The zone named manager has a simple window opening and component. The second zone has a sash window and uses an upper and lower connection to represent this. The third zone has a tall thin window where bi-directional flow is possible and a bi-directional component is used.

In this case we will create an initial model in which the windows are in their open position. Because we may later add control the network includes two connections per opening, one for the window and one for a crack that can act as a bypass in case the window is fully closed. These window and crack combinations are seen in many of the example models. As long as the window is open (even partially) the crack has almost no influence in the solution. If the window closes then the crack ensures that the closed state allows a constrained flow so the solver does not crash. And since windows which have no frame-related infiltration are rare the crack component is a better representation of the flow path than defining a very small orifice.



Basic network with windows

boundary node

internal node

Figure 11.24: network in three test rooms

- Each zone has a window-crack connection to the outside. The crack component (window_crack) is 2.0m x 5mm and is used many times
- Each zone has a door-crack to an adjacent space (which is not geometrically defined). This crack component (door_crack) is 0.8m x 10mm and is used many times
- The adjacent space node should be defined as using the current temperature of the flow node in the first zone.
- Manager has one window opening (window1.67) which is 1.67m2 is defined once and used once
- Manager_t_b (with upper and lower windows) needs 2 windows (win_up_.884 and win_low_.884) which are separately defined.
- Manager_bi (with a bi-directional flow opening). The component (win_bi) is Xm wide, Ym tall, has a discharge coefficient of 0.6 and the distance from its base to the adjacent node is X. Because the distance from the base of the door to the adjacent zone node may differ it is necessary to define difference bi-directional components for use in different contexts.
- Boundary nodes are defined at the specific height of the opening or crack they are associated with. This allows a zero difference in height between the boundary node and the component.

```
   8    6    7    1.000    (nodes, components, connections, wind reduction)
Node          Fld. Type   Height    Temperature    Data_1       Data_2
manager         1    0   1.5000       20.000           0.       40.501
manager_t_b     1    0   1.5000       20.000           0.       40.501
manager_bi      1    0   1.5000       20.000           0.       40.501
gl_ext          1    3   1.9500           0.       5.0000       180.00
low_glz_ext     1    3   1.1500           0.       5.0000       180.00
hi_glz_ext      1    3   2.7500           0.       5.0000       180.00
bi_glz          1    3   1.9500           0.       5.0000       180.00
adjacent        1    0  0.50000E-01    manager               0.       40.501
Comp   Type C+ L+ Description
door_crack    120  3  0 Specific air flow crack             m = rho.f(W,L,dP)
    1.00000    5.00000E-03   0.800000
win_1.68      110  2  0 Specific air flow opening           m = rho.f(A,dP)
    1.00000    1.68000
win_low.84    110  2  0 Specific air flow opening           m = rho.f(A,dP)
    1.00000   0.840000
hi_win.84     110  2  0 Specific air flow opening           m = rho.f(A,dP)
    1.00000   0.840000
bi_win        130  5  0 Specific air flow door              m = rho.f(W,H,dP)
    1.00000   0.880000    1.88000   0.600000   0.600000
win_crack     120  3  0 Specific air flow crack             m = rho.f(W,L,dP)
    1.00000    5.00000E-03    2.00000
+Node         dHght   -Node         dHght   Comp     Snod1  Snod2
gl_ext        0.000   manager       0.225   win_1.68
low_glz_ext   0.000   manager_t_b  -0.175   win_low.84
hi_glz_ext    0.000   manager_t_b   0.625   hi_win.84
bi_glz        0.000   manager_bi    0.225   bi_win
adjacent      0.000   manager      -0.725   door_crack
adjacent      0.000   manager_t_b  -0.725   door_crack
adjacent      0.000   manager_bi   -0.725   door_crack
```

Figure 11.25: ESP-r network file for window opening model

Nodes can be automatically generated for each zone; data is inferred from the zone (volume, reference height, etc). Internal nodes are usually at an unknown pressure. Define the boundary nodes to reflect the position of the openings in the facade.

Ensure your component names match the sketch! A component can be used several places. If control is to be applied - unique names of components and some duplication of components can be helpful (e.g. win_low.84 and hi_win.84).

Connections should tell a story! Parallel connections e.g. an opening and a crack - are useful if control is to be applied.

Hint: mark you sketch as connections are made.

Figure 11.25 is a copy of the completed file which includes the attributes of the nodes and the components and connections. When you are working on the network refer back to this listing.

11.10.1 Component selection

Which window definition works best?

- an air flow opening does one-way flow only, so in cases of single-sided ventilation, flow can be restricted;
- a bi-directional component is intended for doors Ð care is needed in using it in locations such as windows;
- bi-directional flow can be approximated with a pair of air flow openings. Stack effects are accounted for if heights are correctly defined;

None of the available components takes into account high-frequency pressure changes which are one driving force in single-sided ventilation. When an assessment is carried out with this model, the simple window opening in the zone manager results in almost no flow. This is an artefact of this component type - flow is only supported in one direction at each time step. The limiting component is the opening under the door. The flow rates predicted for the sash window and the bi-directional flow component are more in line with expectations.

This does not imply that simple flow components should not be used. Only that they are a poor representation in the case of single sided ventilation.

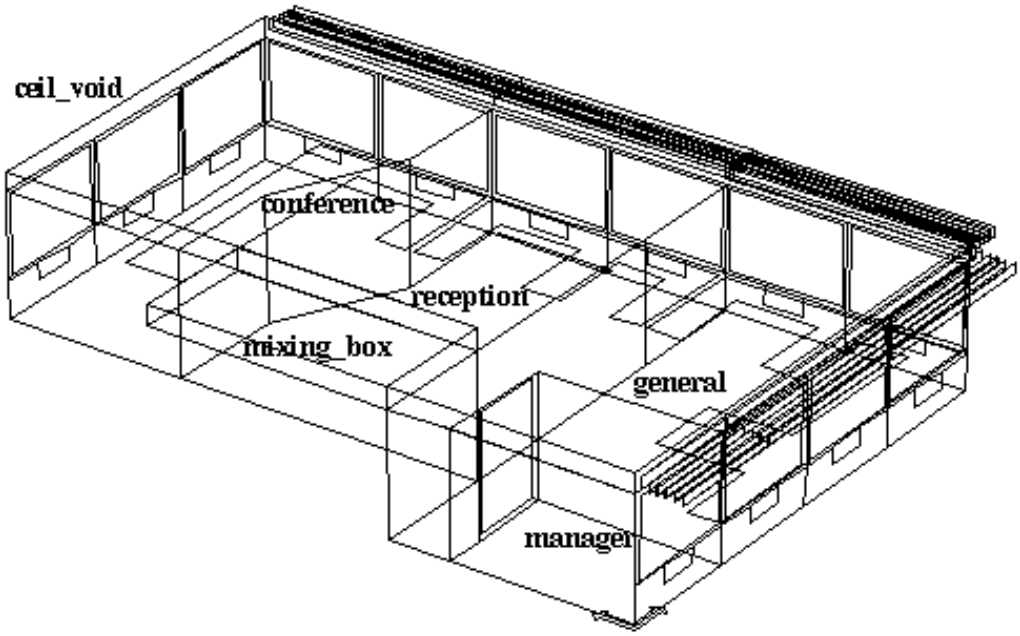


Figure 11.26: portion of an office building

11.11 Schedules vs networks

This section of the *Cookbook* is work-in-progress.

Now we turn to a practical application of a network within a portion of an office building (which includes a reception, conference room, general open-plan office and cellular office). Except for the cellular office, each of the spaces are substantially open to each other (the conference room is only occasionally closed). The facade is an older design and is assumed to be somewhat leaky.

In terms of learning about air flow networks, the design is a good candidate for exploring options for conditioning of the space, including forced and natural ventilation.

The client observation that there are many hours when outside conditions are suitable for mechanical ventilation rather than air conditioning. Are these conditions also suitable for mechanical dampers embedded within the facade to allow fan-free ventilation?

The other feature of this design is the treatment of mixed open plan and cellular spaces. Many simulation teams and some simulation tools *pretend* that there is no air movement between perimeter and core spaces or across open plan spaces. This isolation of the perimeter discounts air transport which allows under heated spaces to borrow heat and spaces that are slightly under-capacity for cooling to borrow cooling from adjacent spaces.

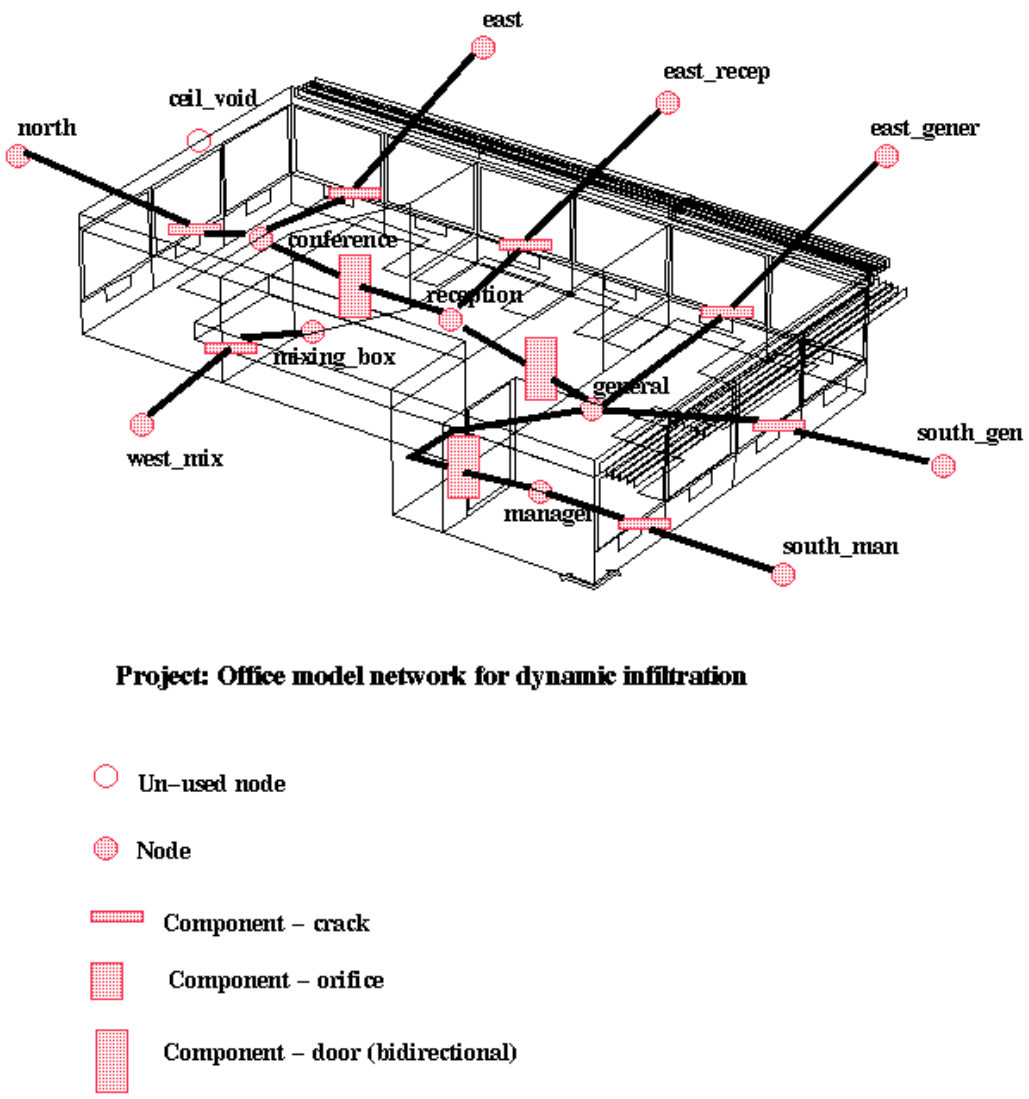


Figure 11.27: infiltration model within office building

In evaluating whether a flow network rather than a flow schedule is appropriate:

- there can be considerable differences between imposed infiltration and the predicted infiltration rates.
- in an open plan office there are large inter-zone flows, resulting in heating/cooling being 'borrowed' from adjacent spaces.

There are three stages to investigate

- the model with scheduled flows and the assumption that there is no intra-zone air movement
- a model variant with zone-to-zone linkages and infiltration paths (Figure 11.27)
- a model variant with zone-to-zone linkages, infiltration paths and controlled dampers on each facade orientation (Figure 11.28).

These three variants are included as example models. If you are interested use these models to explore flow issues. Review the model documentation, especially the sketchs of the flow networks. Compare this with the contents of the flow network file and the interface.

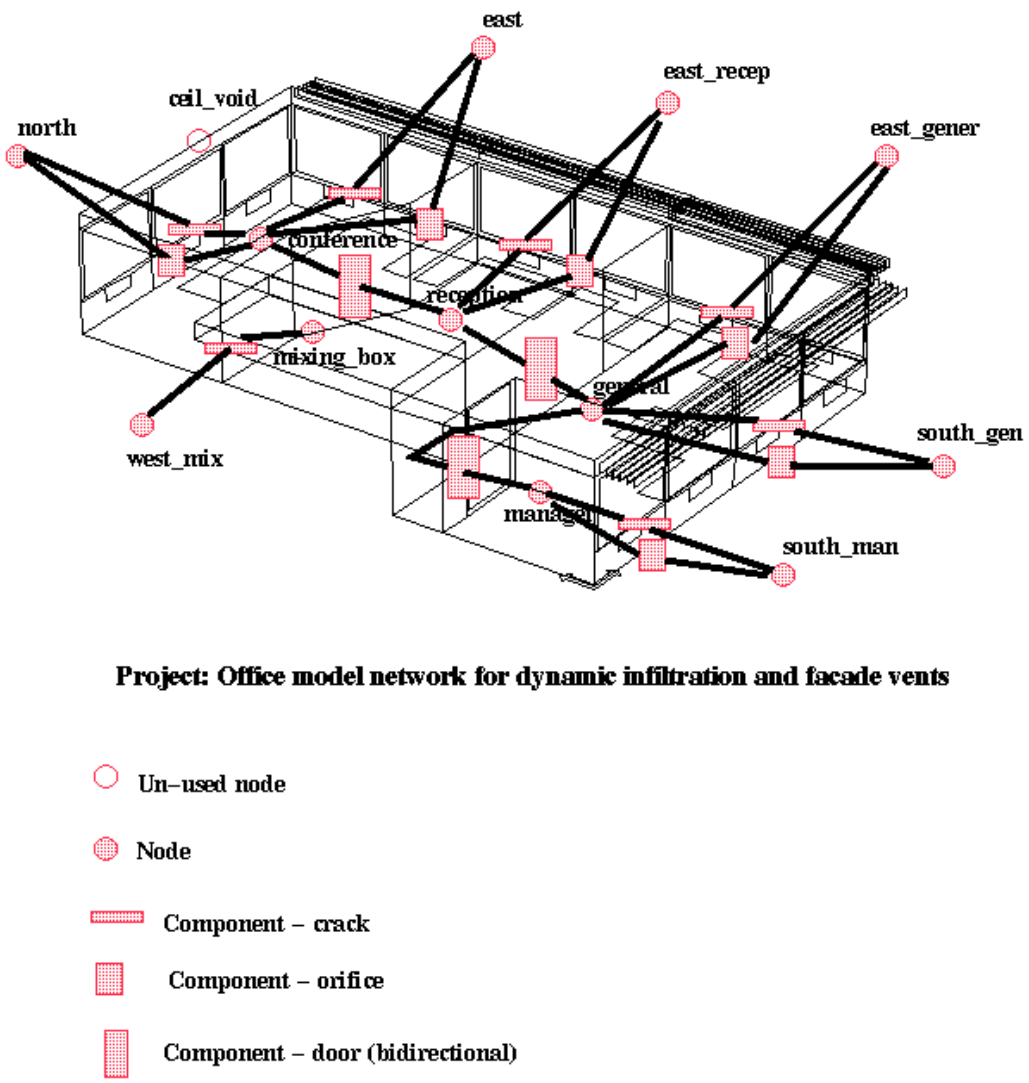


Figure 11.28: vent model within office building

Scheduled flows in an open plan office

The version of the model with scheduled flows will yield predictions where the energy implication of infiltration is dependant on the temperature difference and independent of wind speed, wind direction and facade position. Predictions will follow the pattern seen in Figure 11.29.

Infiltration and inter-zone flow

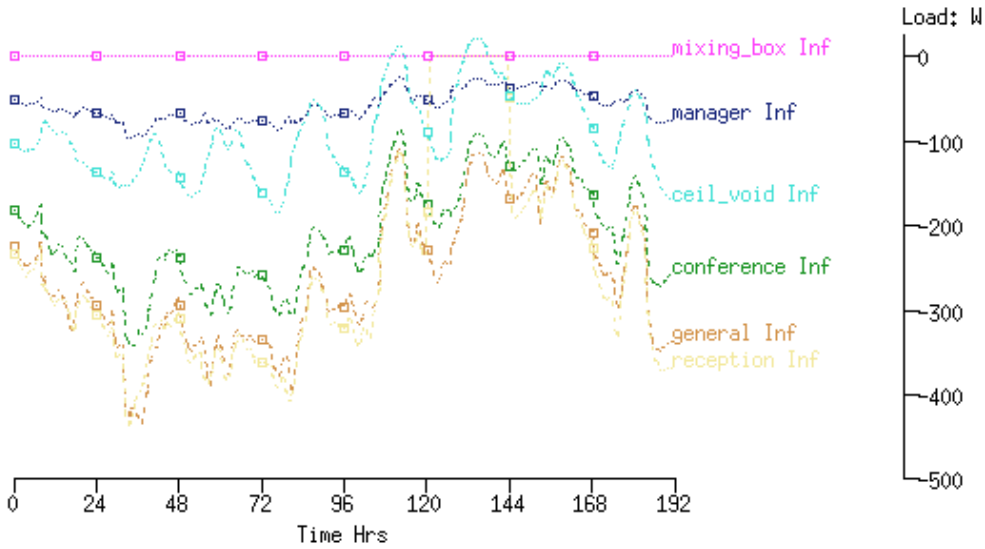
The model variant with infiltration and flow between zones transforms the predictions in several ways (as seen in Figure 11.30): wind speed and direction are now taken into account and there is a moderation of demands as heating and cooling is distributed between zones.

Controlled vents and inter-zone flow

Using the model variant with controlled vents and flow between zones and running an assessment (as in Figure 11.31) indicates that the vents are open slightly longer than necessary and this has performance implications. The data also indicate that the conference room is over ventilated, perhaps because it has two facade orientations and the cross ventilation is more than is required.

Office model with "scheduled" infiltration – energy implications of imposed infiltration

Lib: office_ventapril.res: Results for office_vent (mid-april)
Period: Tue 8 Apr @00h45 to: Tue 15 Apr @23h45 Year:1997 : sim@ 30m, output@ 30m
Zones: manager general conference reception ceil_void mixing_box



Heating and cooling demands and running hours (per zone and total)

Zone total sensible and latent plant used (kWhrs)

Zone id	name	Sensible heating Energy (kWhrs)	No. of Hr rqd	Sensible cooling Energy (kWhrs)	No. of Hr rqd	Humidification Energy (kWhrs)	No. of Hr rqd	Dehumidification Energy (kWhrs)	No. of Hr rqd
1	manager	8.79	60.0	-14.93	51.0	0.00	0.0	0.00	0.0
2	general	22.00	44.5	-83.67	62.5	0.00	0.0	0.00	0.0
3	conference	26.87	54.5	-53.97	55.5	0.00	0.0	0.00	0.0
4	reception	16.95	44.5	-24.00	49.5	0.00	0.0	0.00	0.0
5	ceil_void	0.00	0.0	0.00	0.0	0.00	0.0	0.00	0.0
6	mixing_box	0.00	0.0	0.00	0.0	0.00	0.0	0.00	0.0
	All	74.61		-176.56		0.00		0.00	

Heating and cooling capacity for each zone and diversified totals

Lib: office_ventapril.res: Results for office_vent (mid-april)
Period: Tue 8 Apr @00h45 to: Tue 15 Apr @23h45 Year:1997 : sim@ 30m, output@ 30m
Zone sensible load (kW)

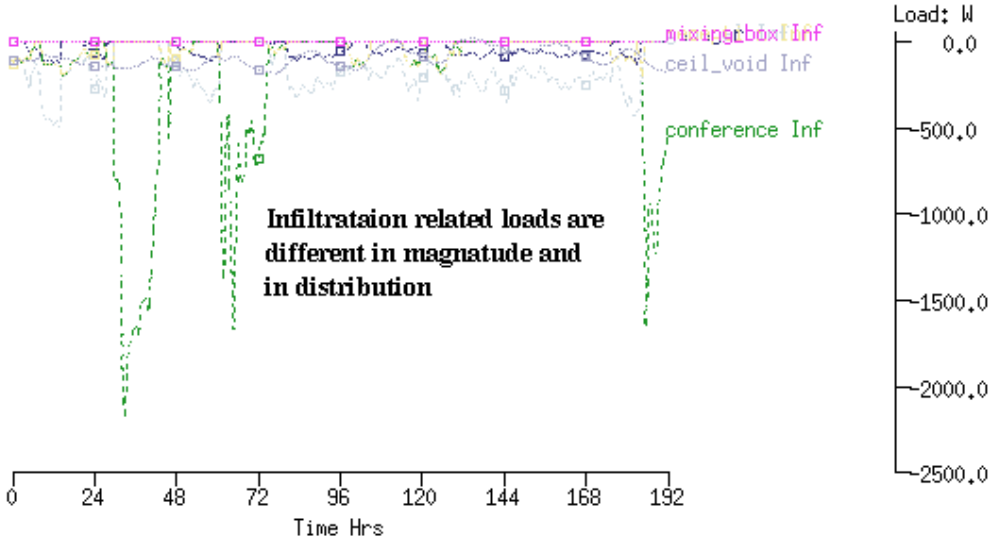
Description	Maximum value	Maximum occurrence	Minimum value	Minimum occurrence	Mean value	Standard deviation
manager	0.42	8 Apr@08h45	-0.65	12 Apr@14h15	-0.03	0.20
general	1.22	9 Apr@08h15	-2.92	15 Apr@14h15	-0.32	0.88
conference	1.38	9 Apr@08h15	-2.60	15 Apr@14h45	-0.14	0.68
reception	1.25	9 Apr@08h15	-1.34	15 Apr@14h15	-0.04	0.38
ceil_void	0.00	8 Apr@00h15	0.00	8 Apr@00h15	0.00	0.00
mixing_box	0.00	8 Apr@00h15	0.00	8 Apr@00h15	0.00	0.00
All	4.25	9 Apr@08h15	-7.34	15 Apr@14h45		

Figure 11.29: scheduled flow performance

Take your time when exploring flow results. Discovering approaches to the use of the results analysis module which yield clear indications of flow performance is an investment well worth making. The quantity of information can be large and there are several different *views* of the flow prediction data as well as reports on the energy implications of flow.

Office model with dynamic infiltration and interzone flows – energy implications of infiltration

Lib: office_vent_netinf_apr: Results for office_vent_netinf
Period: Tue 8 Apr @00h52 to: Tue 15 Apr @23h52 Year:1997 : sim@ 15m, output@ 15m
Zones: manager general conference reception ceil_void mixing_box



Heating use reduced, cooling slightly increased.

Period: Tue 8 Apr @00h52 to: Tue 15 Apr @23h52 Year:1997 : sim@ 15m, output@ 15m

Zone total sensible and latent plant used (kWhrs)

Zone id	Zone name	Sensible heating Energy (kWhrs)	Sensible heating No. of Hr rqd	Sensible cooling Energy (kWhrs)	Sensible cooling No. of Hr rqd	Humidification Energy (kWhrs)	Humidification No. of Hr rqd	Dehumidification Energy (kWhrs)	Dehumidification No. of Hr rqd
1	manager	6.00	41.8	-15.27	53.0	0.00	0.0	0.00	0.0
2	general	15.09	32.2	-84.16	57.2	0.00	0.0	0.00	0.0
3	conference	24.09	41.8	-59.24	50.8	0.00	0.0	0.00	0.0
4	reception	6.43	26.0	-34.14	51.5	0.00	0.0	0.00	0.0
5	ceil_void	0.00	0.0	0.00	0.0	0.00	0.0	0.00	0.0
6	mixing_box	0.00	0.0	0.00	0.0	0.00	0.0	0.00	0.0
	All	51.60		-192.81		0.00		0.00	

Heating capacity increased while cooling capacity the same

Lib: office_vent_netinf_apr: Results for office_vent_netinf
Period: Tue 8 Apr @00h52 to: Tue 15 Apr @23h52 Year:1997 : sim@ 15m, output@ 15m
Zone sensible load (kW)

Description	Maximum value	Maximum occurrence	Minimum value	Minimum occurrence	Mean value	Standard deviation
manager	0.52	8 Apr@08h37	-0.66	12 Apr@14h22	-0.05	0.19
general	1.65	8 Apr@08h07	-2.81	14 Apr@14h22	-0.36	0.86
conference	2.89	9 Apr@08h22	-2.80	15 Apr@14h52	-0.18	0.77
reception	1.52	8 Apr@08h07	-1.54	15 Apr@14h52	-0.14	0.39
ceil_void	0.00	8 Apr@00h07	0.00	8 Apr@00h07	0.00	0.00
mixing_box	0.00	8 Apr@00h07	0.00	8 Apr@00h07	0.00	0.00
All	6.24	9 Apr@08h07	-7.51	15 Apr@14h37		

Figure 11.30: predicted infiltration performance

There is one substantial omission in the reporting of flow. It is not at all easy to get an overview of what is happening throughout the network at a single point in time. Such a feature would speed up the discovery of patterns within a network. Currently users must manually collect this information.

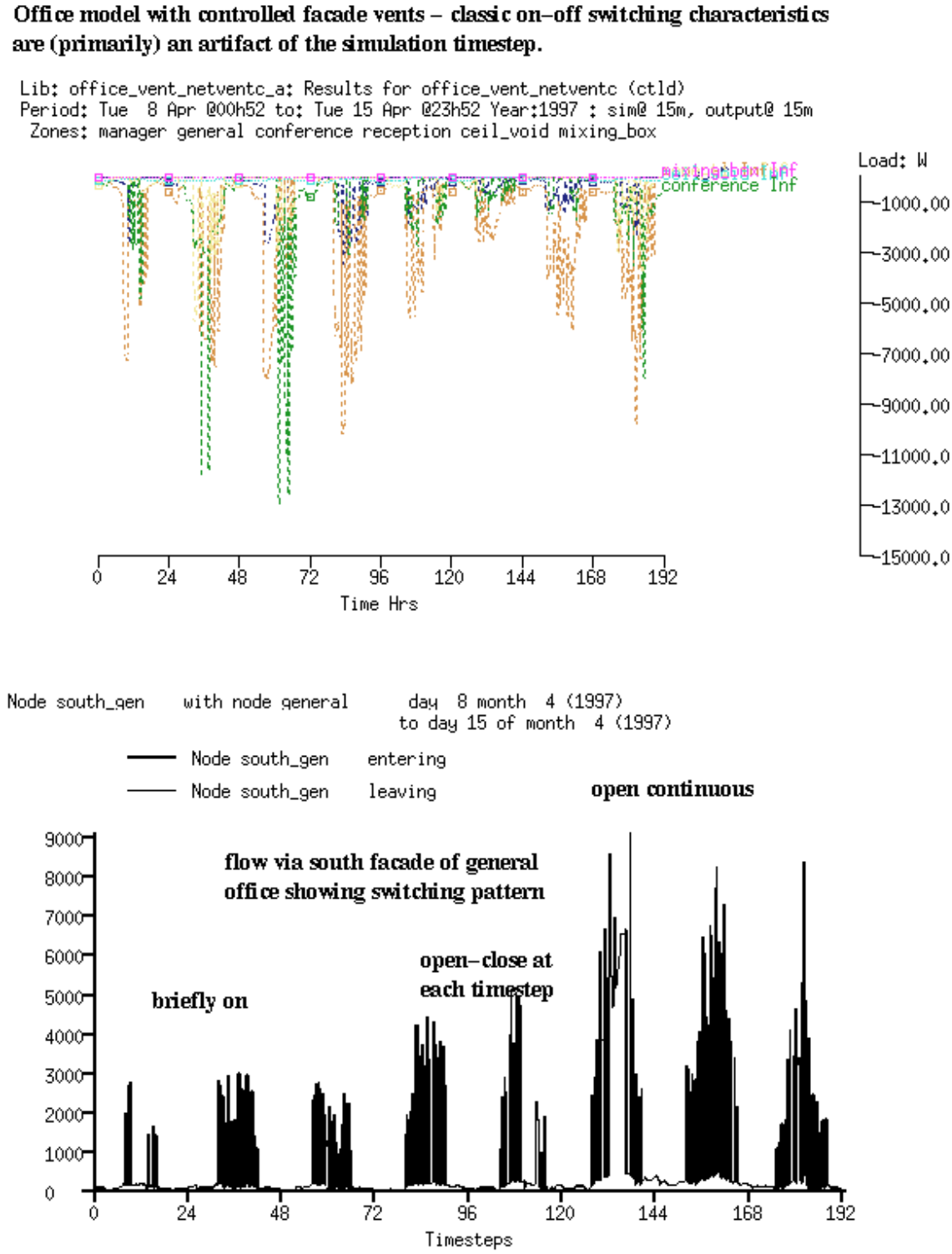


Figure 11.31: predicted facade vent performance

A word about saw-tooth predictions. Flows that oscillate at each time step are sometimes an artefact of the simulation process. If the magnitude of the oscillation is likely to decrease, but there is not time (or disk space) to run assessments at a shorter time step, some users choose to integrate the results when running the simulation. This removes the oscillation but preserves the general trend of flow.

11.12 Limitations of Network Airflow Models

Although network flow models are useful, they are limited for some applications:

- Large volumes represented by a single node, implying well-mixed conditions.
- Temperature distributions within air volumes cannot be determined (e.g. stratification).
- Momentum effects neglected.
- Insufficient resolution for local surface convection determination.

12 Detailed flow via CFD

Given the limitations of mass flow networks mentioned in the previous chapter, research began about a decade ago to extend ESP-r to support higher levels of resolution by incorporating a CFD solver. Although CFD is a mature field of research, its implementation in building models poses a number of classic problems. First, flow velocities in buildings are low (in comparison to traditional applications of CFD) and likely to be within the transition range between flow that is considered turbulent and that which is considered laminar. The second issue is that boundary conditions such as surface temperatures and air temperatures and driving forces change over time.

In buildings the movement of air changes surface temperatures and surface temperature changes alter the flow field. That our virtual physics models do not represent this well has been a considerable irritation to the simulation community.

Conversely, the building solver typically makes rough assumptions about the flow field within the zone and the heat transfer at surfaces. Even in models which include a mass flow network we can only make crude guesses at the velocity that is implied by a given mass transfer.

Clearly, both whole building solvers and CFD solvers would have much to gain by enabling the two solvers to exchange information as the simulation progresses.

What has been implemented in ESP-r is a radical approach to a difficult problem. It assumes that boundary conditions will change so it updates its boundary conditions at each time-step. It assumes that initial directives to the solution process may not be appropriate when conditions change and it re-evaluates the flow field to determine if different so-called *wall functions* should be used. It then creates new directives for the solver to follow to best represent conditions at that timestep and then passes information back to the building solver to use in evaluating heat transfer at the surfaces in the zone. Its default assumption is that the CFD domain is transient rather than static.

The solution also takes into account connections between the CFD domain and a mass flow network. This allows changes in pressure or mass flow in other zones of the model to become new driving forces for the CFD solver. And because mass flow networks also include boundary nodes the CFD domain also has information on changes in weather patterns. And time-varying heat sources within rooms from the zone operations schedules are noticed by the CFD solver and can be

associated with blocks of cells.

Powerful stuff. And also utterly wrapped up in jargon which sounds like English but means something different. CFD is a place where dragons live. ESP-r's interface to CFD has a steep learning curve.

The *Cookbook* is not a tutorial on the theory of CFD or the so-called *conflation mechanisms* used by the solvers. There are several PhD thesis written on the topic which are available for download on the ESRU web site publications page at <<http://www.esru.strath.ac.uk>>. And there are any number of books on the subject. And the source code associated with CFD is heavily commented and can provide a number of useful clues.

If you already have a solid background in CFD and are comfortable with what you found in the literature search mentioned above, then continue reading this chapter. If not, CFD will absorb both computing and mental resources at an painful rate. You have been warned.

The chapter includes an overview of the entities and parameters which can be used to define a CFD domain, methods for designing a gridding scheme within the domain, and what to look for in the performance predictions. There are also some information boxes and dragon boxes where particular care should be exercised.

This chapter will be completed at a later date.

13 Plant

In ESP-r environmental control systems can be represented as either so-called idealised zone/flow controls or as a network of system components which is often called a plant system. The choice of which approach to take is partly based on how much you want to know about the detailed performance of the environmental control system and partly on how much descriptive information you can acquire about the composition of the environmental control.

The *Cookbook* does not cover the theory of component networks or the solution techniques used. It provides an outline strategy for using systems network facilities. Readers might treat this Chapter as an initial draft as there are many dragons lurking with this portion of ESP-r and there remain many gaps in the strategies.

Networks of components offer the following facilities:

- the psychrometric state within components and at points in the network is explicitly computed and is available for inspection
- interactions between components and/or controls are computed at subminutely intervals can be inspected in this time domain
- those who have an interest in fine-tuning the response of particular components or control devices within the network have many options for creating models which are close approximations
- those interested in high resolution of both system components and mass flows can link both the system component solver and the mass flow solver

ESP-r provides feedback on the composition of such networks and a wealth of information about what is happening within and between components during simulations (via trace facilities) and different views of the *state variables* within the **res** module.

Those who master the use of system components are able to address a range of questions that are not possible with other approaches and have access to a rich store of performance indicators.

Tactical users of simulation do not rush to create networks of components until they have learned all they can from ideal controls. And they do this because creating networks of components:

- tends to take longer (more descriptive information and more linkages between components)
- such networks need tuning like real systems

- such networks fail in similar ways to real systems
- some system interactions are in the frequency of seconds or fractions of a second and so the volume of information increases greatly.
- much performance information is in a form that it difficult for many users to interpret
- the facility requires you to assemble a network that is both syntactically correct and physically correct

In comparison with most of the ideal zone controls the use of networks of system components involves a steeper learning curve. Many tasks and much of the quality assurance in models with system components is the responsibility of the users. A methodical approach is essential.

- Rule one: start with zone and/or flow controls learn as much as possible about the pattern of demands and the likely control logic that is appropriate for the design
- Rule two: planning and sketches are essential.
- Rule three: walk before you run - test out portions of the network and control options on a simple model before scaling the network.
- Rule four: document what you do.
- Rule five: leave plenty of time for testing.

For readers who are approaching the use of system networks in ESP-r with prior experience with component based analysis be aware that all components are entities within a single network. In ESP-r there is no conceptual difference between components representing a duct, a valve, or a cooling tower. There is no concept of central plant and zone-side components.

13.1 Using a network to represent mechanical ventilation

Mechanical ventilation is one aspect of building design which simulation can play a role. We will create several models of a mechanical ventilation design to explore such systems respond to changing demands and boundary conditions.

The first approach is to represent all aspects of a mechanical ventilation design within the component network (i.e. it uses a component as a simplified representation of a thermal zone). Although this will not provide a full accounting of the interactions between supply and demand sides it is an approach suited to early design stage questions where little might be known of the zones.

Figure 13.1 shows a standard mechanical ventilation system which has a supply and an exhaust fan and a heater coil just up-stream from the supply fan. Two zones are supplied and the extracts from each zone are combined in a mixing box just before the exhaust fan.

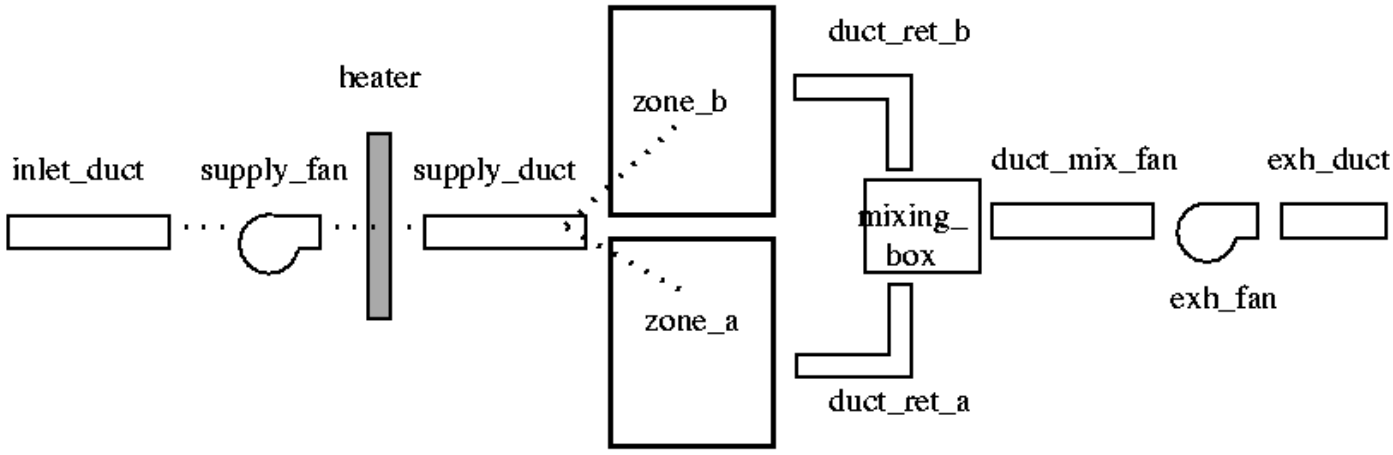


Figure 13.1 Basic mechanical ventilation system.

Planning is essential, even for a simple model. Sketch out your network first and decide on names for the components. Most of your work within the project manager will involve names of components and numbers representing component attributes and your sketch is essential for keeping track of your work, supporting QA tasks and communicating with clients.

After sketching the network gather the component information. The list on the next page contains component information for the 12 components and you should refer to this as you create your network.

Components have a sequence - the initial group goes from the returns from the zones to the exhaust and then come the idealized zones and then the inlet_duct to supply_duct. Adopting a sequence which proceeds from the return to the supply can make subsequent tasks easier.

After the name of the component is a number in ( ) which is the component index within the plant network. Include this number on your sketch in addition to the component name. Why? Because there are a few places in the interface where you have to type in this index rather than selecting from a list of component names.

Most components include an attribute for the total mass of the component. For these exercises this need not be exact. There is also a mass weighted average specific heat which tends to be either 500.00 or 1000.00. Each component also has a UA modulus. These parameters support calculations of how the casing of the component interacts with its surroundings.

Ducts have additional parameters, including the length of the duct, cross-sectional area and hydraulic diameter. If you pre-calculate these it will speed up your descriptive tasks as well as reducing mistakes (see Rule two).

Project Manager: enquiries to esru@strath.ac.uk

```
Cross sectional face area (m^2)                    :  0.12270E-01

Component: duct_ret_b ( 2) code  2, db reference  6
No Control data
Modified parameters for duct_ret_b
Component total mass (kg)                          :   1.8500
Mass weighted average specific heat (J/kgK)        :   500.00
UA modulus (W/K)                                   :   2.8000
Hydraulic diameter of duct (m)                     :  0.12500
Length of duct section (m)                         :   1.0000
Cross sectional face area (m^2)                    :  0.12270E-01

Component: mixing_box ( 3) code  3, db reference  1
No Control data
Modified parameters for mixing_box
Component total mass (kg)                          :   1.0000
Mass weighted average specific heat (J/kgK)        :   500.00
UA modulus (W/K)                                   :   3.5000

Component: duct_mix_fan ( 4) code  4, db reference  6
No Control data
Modified parameters for duct_mix_fan
Component total mass (kg)                          :   9.2500
Mass weighted average specific heat (J/kgK)        :   500.00
UA modulus (W/K)                                   :   14.000
Hydraulic diameter of duct (m)                     :  0.12500
Length of duct section (m)                         :   5.0000
Cross sectional face area (m^2)                    :  0.12270E-01

Component: exh_fan ( 5) code  5, db reference  3
Control data:    0.060
Modified parameters for exh_fan
Component total mass (kg)                          :   10.000
Mass weighted average specific heat (J/kgK)        :   500.00
UA modulus (W/K)                                   :   7.0000
Rated total absorbed power (W)                     :   50.000
Rated volume flow rate (m^3/s)                     :  0.10000
```

Components

	Name	Iref no.	Type
a	duct_ret_a	6	air conditioning
b	duct_ret_b	6	air conditioning
c	mixing_box	1	air conditioning
d	duct_mix_fan	6	air conditioning
e	exh_fan	3	air conditioning
f	exh_duct	6	air conditioning
g	zone_a	25	solar and other
h	zone_b	25	solar and other
i	inlet_duct	6	air conditioning
j	supply_fan	3	air conditioning
k	heater	5	air conditioning
l	supply_duct	6	air conditioning

\+ Add/Delete/Copy

? Help
\- Exit

Figure 13.2 Components (with attributes shown).

Connections

	Sending comp	@ Node		to	Receiving comp	@ Node	Conn Type	Mass Div
a	outside air	ambient		-->	inlet_duct	air node 1	zone/amb	1.000
b	inlet_duct	air node	1	-->	supply_fan	air node 1	to compt	1.000
c	supply_fan	air node	1	-->	heater	air node 1	to compt	1.000
d	heater	air node	1	-->	supply_duct	air node 1	to compt	1.000
e	supply_duct	air node	1	-->	zone_a	air node 2	to compt	0.500
f	supply_duct	air node	1	-->	zone_b	air node 2	to compt	0.500
g	zone_a	air node	2	-->	duct_ret_a	air node 1	to compt	1.000
h	zone_b	air node	2	-->	duct_ret_b	air node 1	to compt	1.000
i	duct_ret_a	air node	1	-->	mixing_box	air node 1	to compt	1.000
j	duct_ret_b	air node	1	-->	mixing_box	air node 1	to compt	1.000
k	mixing_box	air node	1	-->	duct_mix_fan	air node 1	to compt	1.000
l	duct_mix_fan	air node	1	-->	exh_fan	air node 1	to compt	1.000
m	exh_fan	air node	1	-->	exh_duct	air node 1	to compt	1.000

\+ add/delete/copy

Figure 13.3 Connections between components.

The pattern for creating components is similar (see Figure 13.5). When you have finished defining the components you should see something like Figure 13.4. Save your network and ake a moment to review the components listed in the interface menu with your sketch to ensure they are consistent.

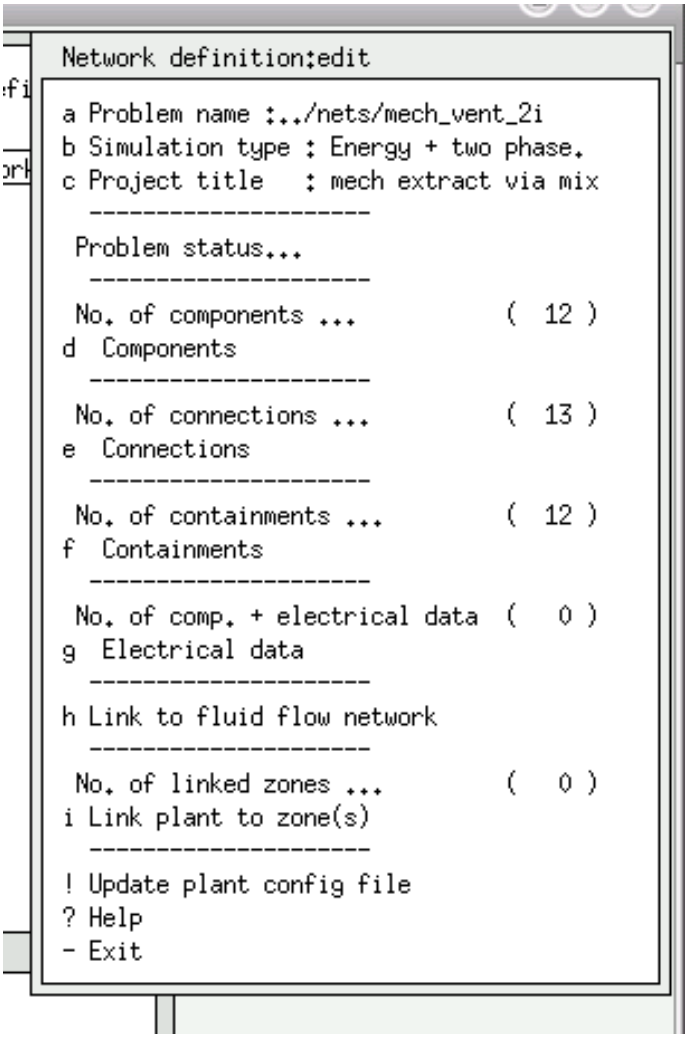


Figure 13.4 Finished network.

The next task is to link the components together. Linking plant components together is different from linking flow network components together. Clear your mind - the pattern is to begin your focus at a component that receives flow and figure out which component is sending the flow. Referring back to Figure 13.1 - the heater is supplied by the supply_fan so when you set up a link the first component in the link is the heater and the second component in the link is the supply_fan.

Look again at the list in Figure 13.3 and draw this (a coloured line works well) on your sketch of the network. When this makes sense, start the process of adding connections and noting them on your sketch. The mass diversion for `supply_duct ->zone_a` and `supply duct ->zone_b` are 0.5 because each takes half of the output of the component supply_duct. Except for the receiving component inlet_duct, which takes its supply from ambient air, each of the other connection is with another component. When the connections are complete save the network.

```
Component: duct_ret_a ( 1)
db reference  6
Modified parameters for duct_ret_a
Component total mass (kg)                  :  3.7000
Mass weighted average specific heat (J/kgK): 500.00
UA modulus (W/K) :  5.6000
Hydraulic diameter of duct (m)             :  0.12500
Length of duct section (m)                 :  2.0000
Cross sectional face area (m^2)          : 0.12270E-01

Component: duct_ret_b ( 2) db reference  6
Modified parameters for duct_ret_b
Component total mass (kg)                  :  1.8500
Mass weighted average specific heat (J/kgK): 500.00
UA modulus (W/K)                           :  2.8000
Hydraulic diameter of duct (m)             :  0.12500
Length of duct section (m)                 :  1.0000
Cross sectional face area (m^2)          : 0.12270E-01

Component: mixing_box ( 3) db reference  1
Modified parameters for mixing_box
Component total mass (kg)                  :  1.0000
Mass weighted average specific heat (J/kgK):  500.00
UA modulus (W/K)                           :  3.5000

Component: duct_mix_fan ( 4) db reference  6
Modified parameters for duct_mix_fan
Component total mass (kg)                  :  9.2500
Mass weighted average specific heat (J/kgK):  500.00
UA modulus (W/K)                           :  14.000
Hydraulic diameter of duct (m)             : 0.12500
Length of duct section (m)                 :  5.0000
Cross sectional face area (m^2)          : 0.12270E-01

Component: exh_fan ( 5) db reference  3
Control data:    0.060
Modified parameters for exh_fan
Component total mass (kg)                  :  10.000
Mass weighted average specific heat (J/kgK):  500.00
UA modulus (W/K)                           :  7.0000
Rated total absorbed power (W)             :  50.000
Rated volume flow rate (m^3/s)             : 0.10000
Overall efficiency (-)                     : 0.70000

Component: exh_duct ( 6) db reference  6
Modified parameters for exh_duct
Component total mass (kg)                  :  5.5000
Mass weighted average specific heat (J/kgK):   500
UA modulus (W/K)                           :  8.4000
Hydraulic diameter of duct (m)             :  0.12500
Length of duct section (m)                 :  3.0000
Cross sectional face area (m^2)          : 0.12270E-01

Component: zone_a ( 7) db reference 25
Control data: -500.000
Modified parameters for zone_a
Component total mass (kg)                  :  10920.
Mass weighted average specific heat (J/kgK):  1000.0
Wall U value (W/m^2K)                      :  0.40000
Total surface area of walls (m^2)          :  78.000
Zone space volume (m^3)                    :  45.000
Inside heat transfer coefficient (W/m^2K)  :  5.0000
Outside heat transfer coefficient (W/m^2K) :  18.000
Outside air infiltration (ACH)             :  0.0000

Component: zone_b ( 8) db reference 25
```

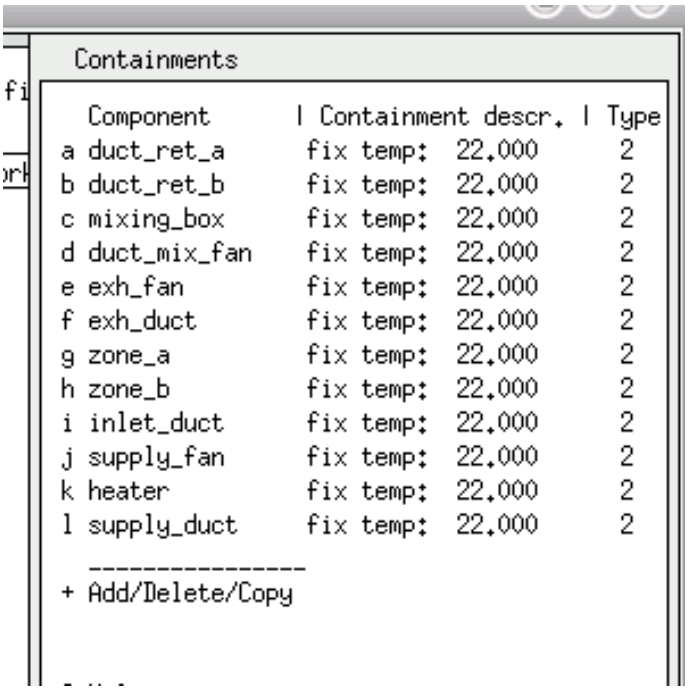
```
Control data:-1000.000
Modified parameters for zone_b
Component total mass (kg)                  :  7560.0
Mass weighted average specific heat (J/kgK):  1000.0
Wall U value (W/m^2K)                      : 0.40000
Total surface area of walls (m^2)          :  54.000
Zone space volume (m^3)                    :  27.000
Inside heat transfer coefficient (W/m^2K)  :  5.0000
Outside heat transfer coefficient (W/m^2K) :  18.000
Outside air infiltration (ACH)             :  0.0000

Component: inlet_duct ( 9) db reference  6
Modified parameters for inlet_duct
Component total mass (kg)                  :  9.2500
Mass weighted average specific heat (J/kgK):  500.00
UA modulus (W/K)                           :   14.000
Hydraulic diameter of duct (m)             :  0.12500
Length of duct section (m)                 :  5.0000
Cross sectional face area (m^2)        :  0.12270E-01

Component: supply_fan (10) db reference  3
Control data:    0.060
Modified parameters for supply_fan
Component total mass (kg)                  :   10.000
Mass weighted average specific heat (J/kgK):   500.00
UA modulus (W/K)                           :   7.0000
Rated total absorbed power (W)             :   50.000
Rated volume flow rate (m^3/s)             :  0.10000
Overall efficiency (-)                     :  0.70000

Component: heater (11) db reference  5
Control data: 3000.000
Modified parameters for heater
Component total mass (kg)                  :   15.00
Mass weighted average specific heat (J/kgK):   1000.0
UA modulus (W/K)                           :   3.5000

Component: supply_duct (12) db reference  6
Modified parameters for supply_duct
Component total mass (kg)                  :   1.8500
Mass weighted average specific heat (J/kgK):   500.00
UA modulus (W/K)                           :   2.8000
Hydraulic diameter of duct (m)             :  0.12500
Length of duct section (m)                 :   1.0000
Cross sectional face area (m^2)         : 0.12270E-01
```

```
Plant comp : duct_ret_a

a Component total mass (kg)                      :  3.7000
b Mass weighted average specific heat (J/kgK)    :  500.00
c UA modulus (W/K)                               :  5.6000
d Hydraulic diameter of duct (m)                 : 0.12500
e Length of duct section (m)                     :  2.0000
f Cross sectional face area (m^2)                : 0.12270E-01
* All items in list
  ---------------------------------------------

? Help
- Exit
```

Figure 13.5 Typical component menu.

13.2 Defining containments

Components exist in a context (or containment), such as surrounded by a fixed or ambient temperatures. For purposes of this exercise we want to attribute each component with a fixed temperature of 22°C. Figure 13.6 shows what you should expect to see.



```
Containments

  Component        | Containment descr. | Type
a duct_ret_a        fix temp:  22.000      2
b duct_ret_b        fix temp:  22.000      2
c mixing_box        fix temp:  22.000      2
d duct_mix_fan      fix temp:  22.000      2
e exh_fan           fix temp:  22.000      2
f exh_duct          fix temp:  22.000      2
g zone_a            fix temp:  22.000      2
h zone_b            fix temp:  22.000      2
i inlet_duct        fix temp:  22.000      2
j supply_fan        fix temp:  22.000      2
k heater            fix temp:  22.000      2
l supply_duct       fix temp:  22.000      2
  ----------------
+ Add/Delete/Copy
```

Figure 13.6 Containments for each component.

7.3 Finishing off the model and testing

At this point your interface should look like Figure 13.4. Notice that there is a place for you to include notes (Rule 4) before you forget what this network is about!

Next we need to test the model to see if it is complete and syntactically correct. In the simulator interface look for the `Simulation parameters` option and provide the name of the results file, the period of the simulation and what sort of timestep to use. This allows you to re-run this assessment without having to look around for scraps of paper.

The parameters shown in Figure 13.7 are a good starting point. Once these are set commission an interactive simulation.

The simulator will notice that the model includes only a network of components and will solve only a system only simulation. It will request confirmation for using zero startup days (accept this), the default climate and the name of the results file to be created (write this name down, you will need it in a few minutes). The simulation should take a few seconds. Exit from the simulator and invoke the results analysis module.

> Note for some versions of ESP-r the initial results file name in the results analysis module is incorrect and needs to be edited.

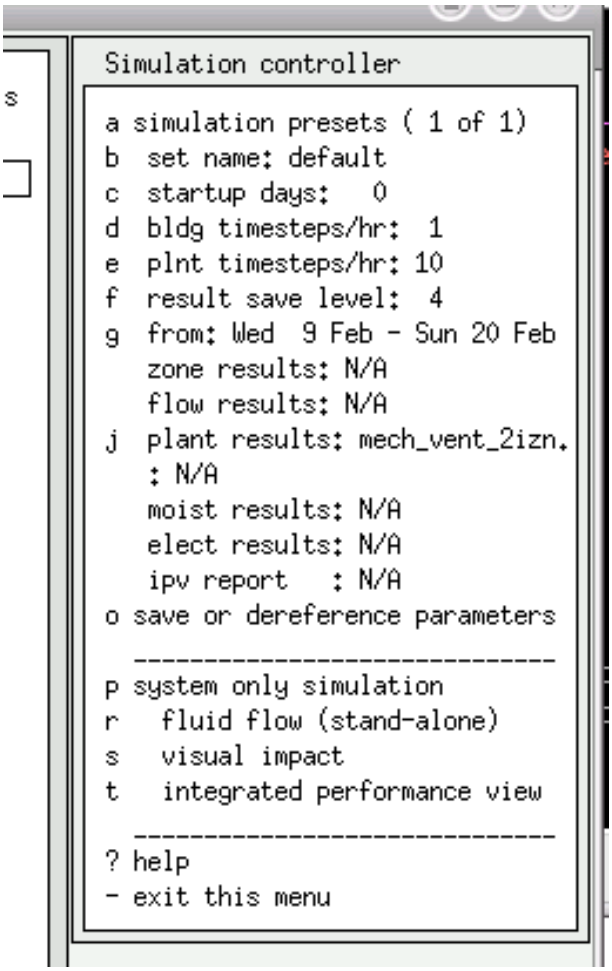


Figure 13.7 Simulation parameters.

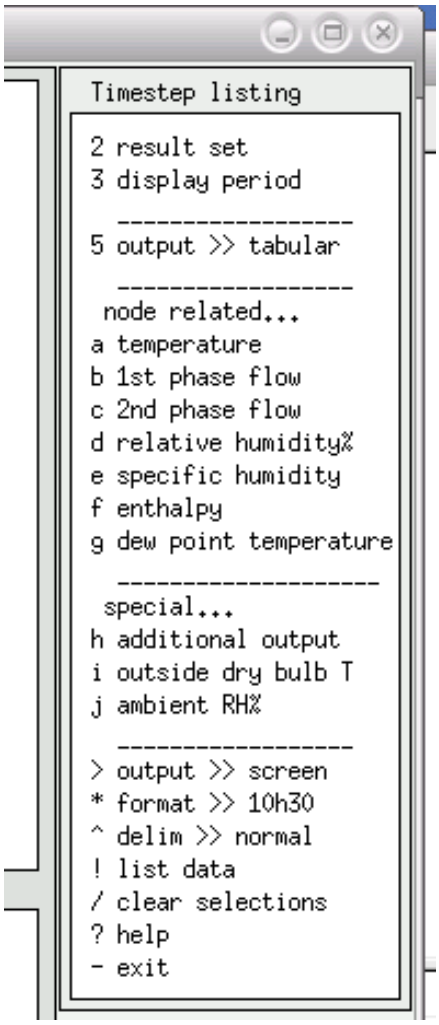


Figure 13.8 Results analysis menu.

The results analysis facilities (see Figure 13.8) for network components allows you to toggle between tabular, psychrometric chart, summary statistics, histograms and graphic plots. Items selected will be re-displayed as you switch views.

Figure 13.9 shows the temperatures at return ducts a & b and Figure 13.10 statistics of temperature and enthalpy at return_duct_b.

Spend a few moments browsing the reports and graphs in search of patterns that indicate how the ventilation system is working.

The diversion ratio of 0.5 from the supply duct to zone_a and zone_b results in the return from zone_b being cooler than from zone_a. Edit the diversion ratios and see if the differences in temperature might be reduced.

To save time, note the information for the two connections before starting the edit. And remember to save the component network file to a slightly different name when you make such changes so that you recover the original file. After saving the changes and commissioning another simulation check and see the change in performance (Figure 13.11).

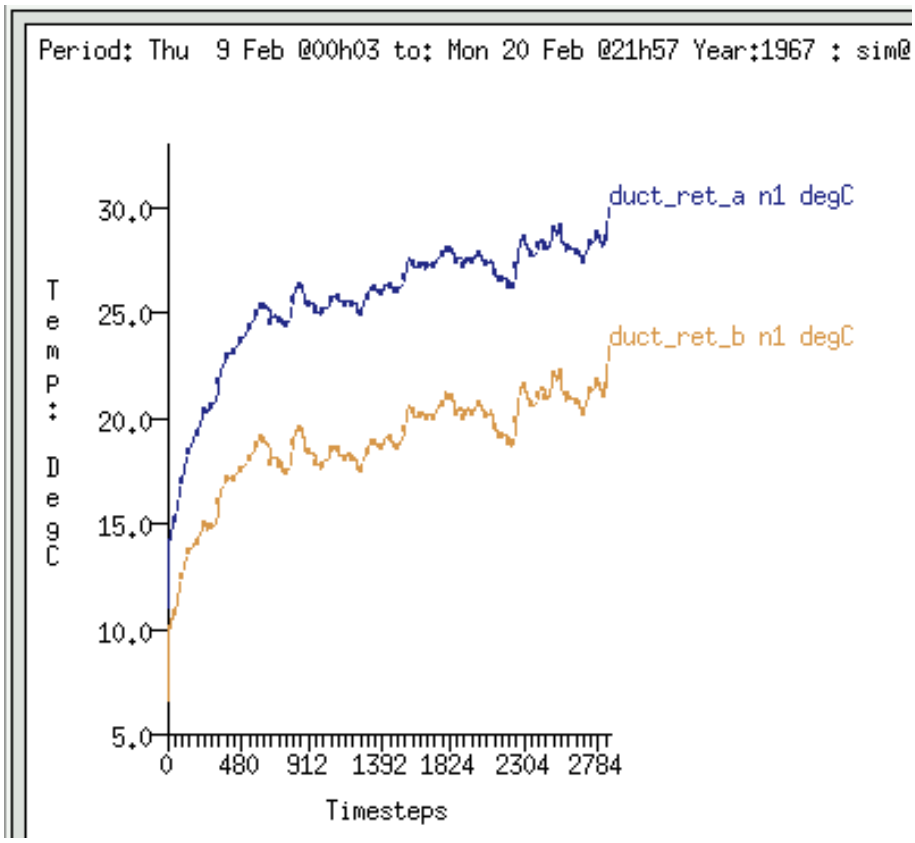


Figure 13.9 Graph of return duct temperature.

capture

Selected duct_ret_b n1 degC
Selected duct_ret_b n1 KJ/Kg

	Maximum		Minimum		Mea
	value	occurrence	value	occurrence	val
duct_ret_b n1 degC	26.56	20 Feb@21h46	13.16	9 Feb@01h50	21.7
duct_ret_b n1 KJ/Kg	45.92	20 Feb@21h46	21.64	9 Feb@01h50	31.5

capture text

Figure 13.10 Statistics at return duct b.

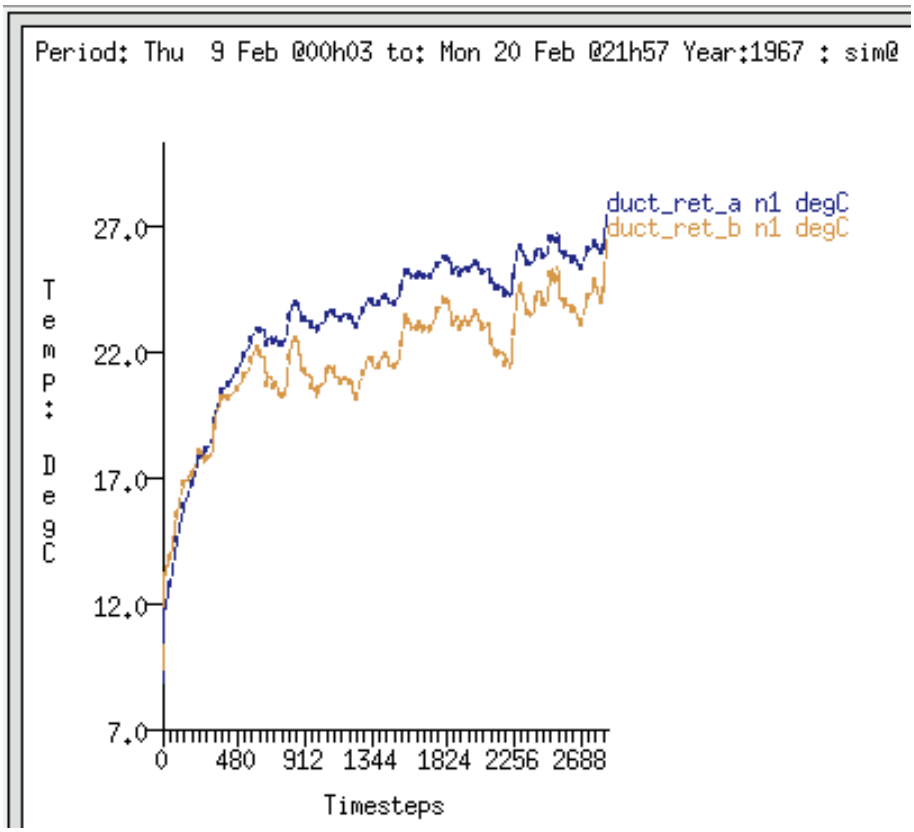


Figure 13.11 Temperatures in return ducts after adjustment.

13.3 Moving from ideal demands to thermal zone demands

In the initial model demands that the mechanical ventilation had to respond to were via component representations of zones. You can also associate a network of environmental system components with thermal zones. In this case we need to add two zones to the model which will have the same volume and surface area and overall thermophysical properties as the component representations.

The component zone_a has a volume of 45m^3 and a zone which is 4m wide x 4m deep x 2.81m high will be equivalent in volume and surface area. The component zone_b has a volume of 27m^3 and a zone which is 4m wide x 2.4m deep x 2.81m high will be equivalent. If all surfaces in the rooms are attributed with the construction external_wall and face the outside then the overall UA will be similar to that of the component representation. These zones are rectangular and have no windows or doors, so the process of creating the geometry and applying attribution is straightforward.

One option is to begin a new model and to build up both the zone and component network to match the requirements of the exercise. A second option would be to upgrade the existing model to include the zones and to adapt the existing network of components. Both options have benefits and drawbacks and it is well worth exploring both.

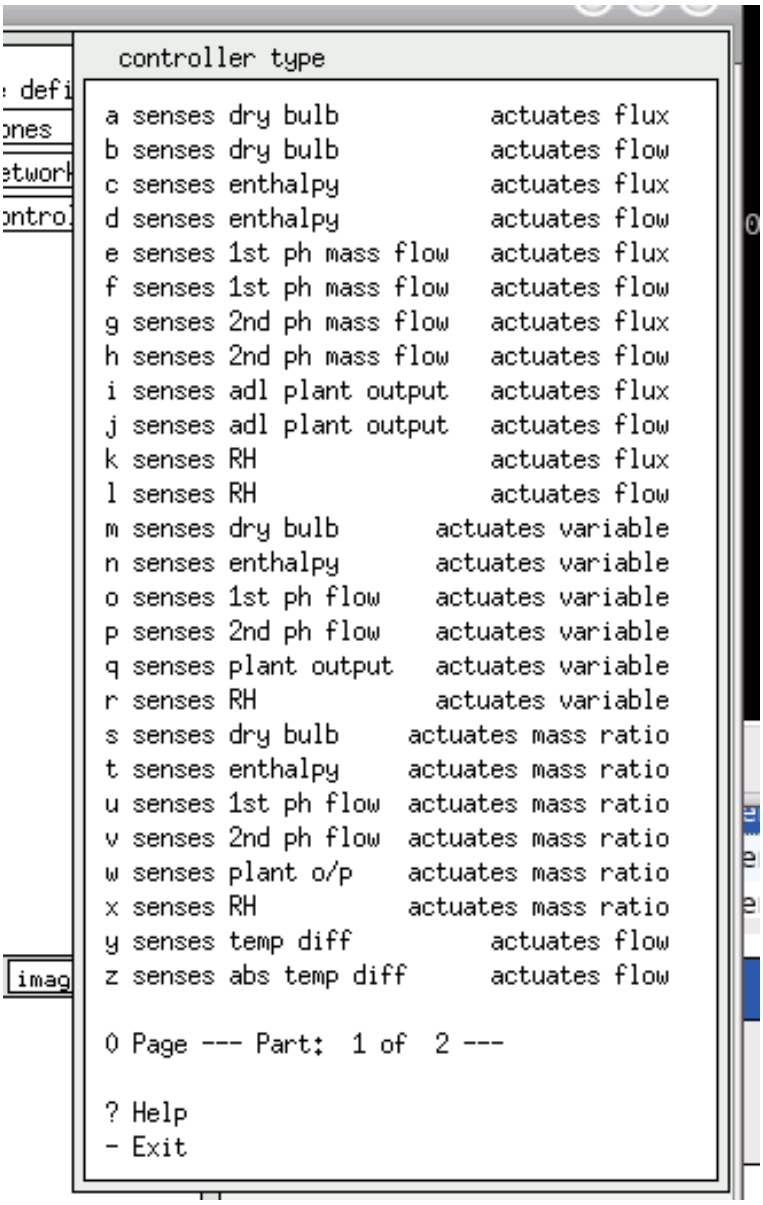


Figure 13.12 Component control types.

For this exercise lets modify the existing model and the first task is to make a backup copy of the model. If the original model folder is named mech_vent then give the following command:

```
cp -r mech_vent
mech_vent_2z
```

Then go into the configuration folder of mech_vent_2z and restart the project manager with the configuration file we want to modify. As soon as the model loads change the root name to mech_vent_2z and alter the model description phrase (as a reminder that this is a different model).

The model has no zones, so go to the zone composition and create zone_a and then zone_b based on the information given above. At the end of this process you would see something like Figure 13.13 for zone_a and something like Figure 13.14 for zone_b.

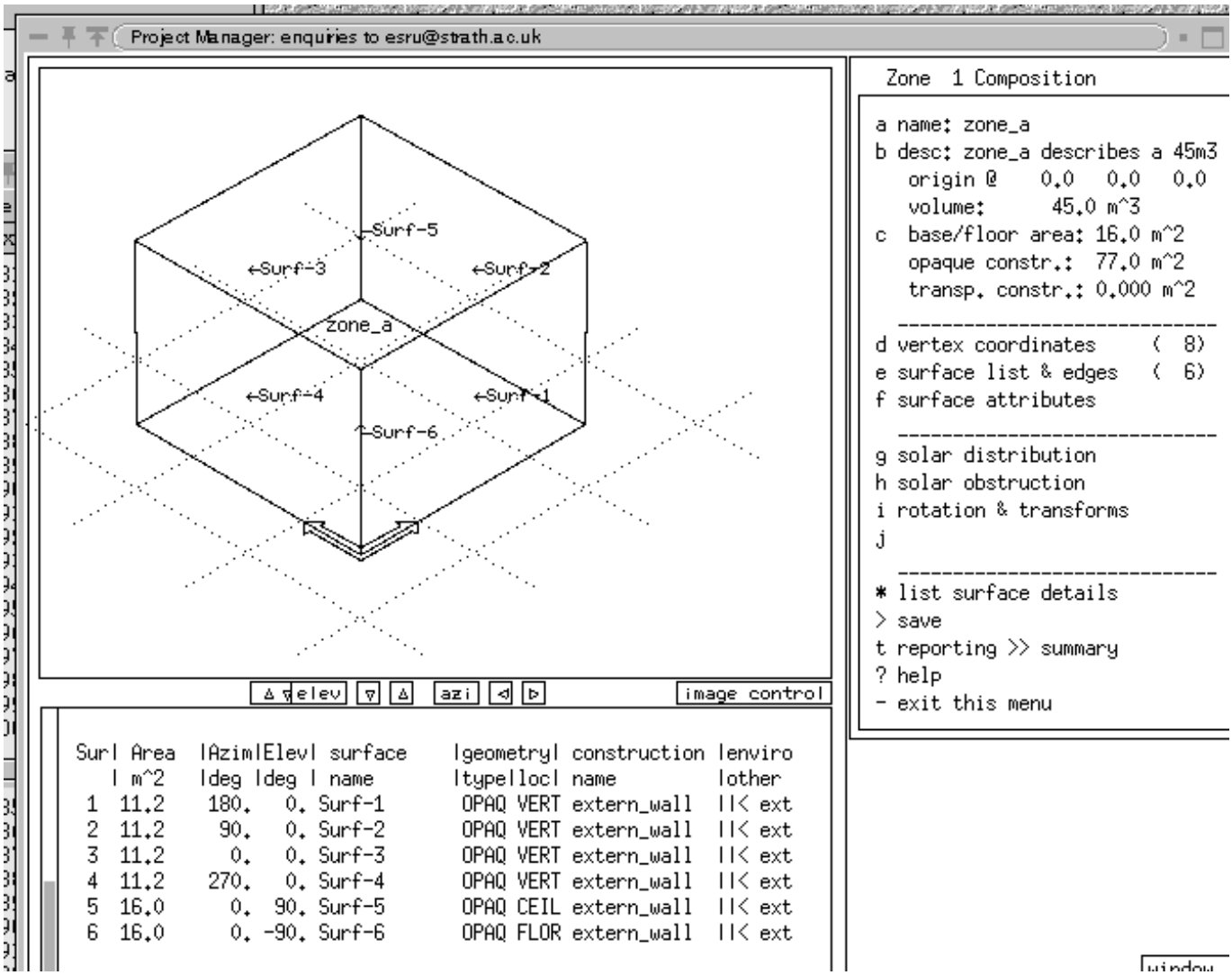


Figure 13.13 Zone a details.

Schedules and other attributes

The two zones need to be fully attributed in terms of composition and operational details. Keep these descriptions simple - 200W during office hours and 0.2 air changes of infiltration will suffice.

Modifying existing network of components takes several steps: first make a backup copy of the existing network, second change the connection to supply_duct -> zone_a (a connection between components) to supply_duct -> duct_ret_a (a connection between a component and thermal zone_a).

In this case the receiving component becomes duct_ret_a, the connection type is 'from a building zone' and then zone_a is selected from the list of available zones. The next question is about the supply for the zone and this remains component supply_duct with a diversion ratio of 0.5.

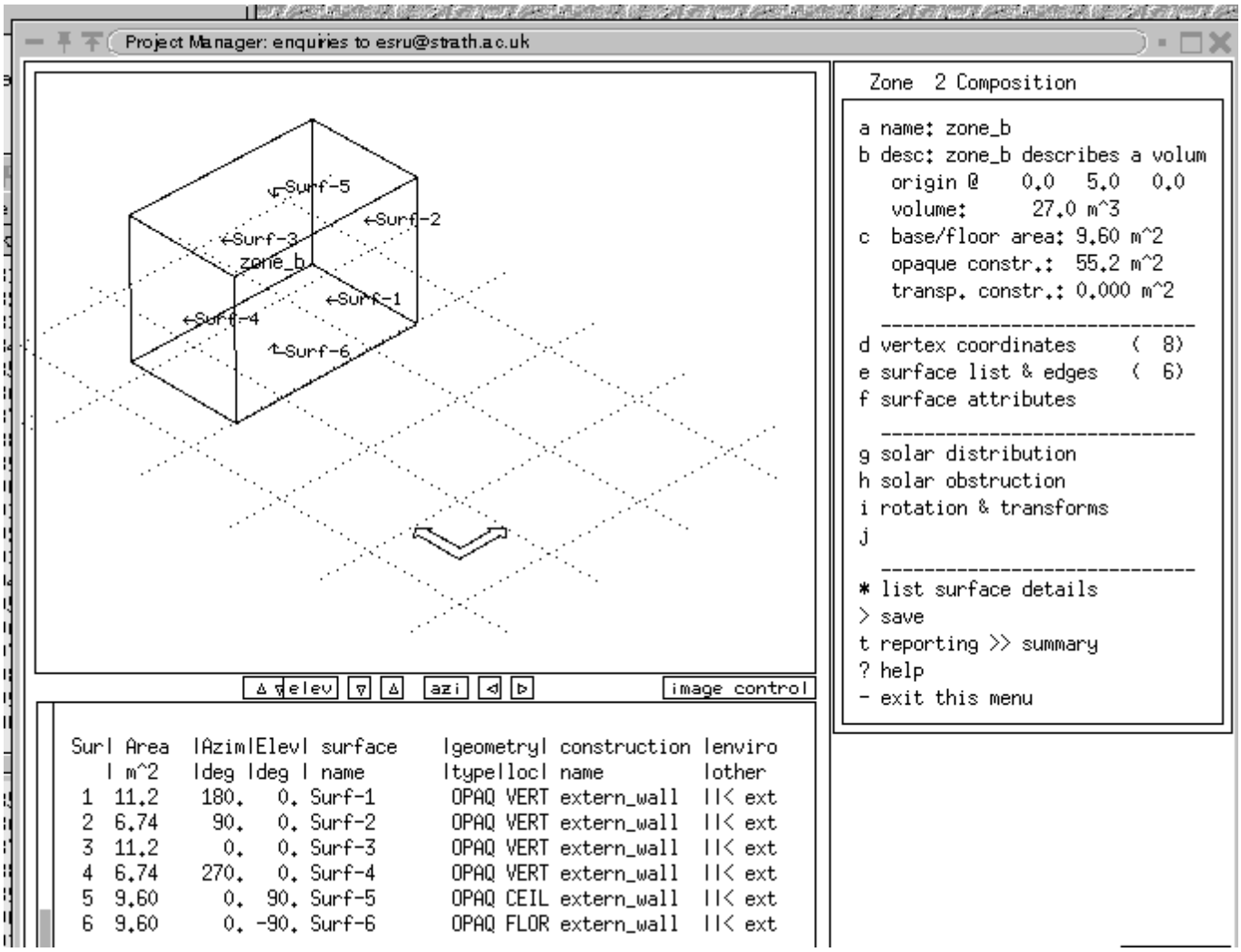


Figure 13.14 Zone b details.

The same needs to be done with the connection supply_duct -> zone_b to become supply_duct -> duct_ret_b (a connection between a component and thermal zone_b). After these changes have been made the interface will look like Figure 13.15.

The next step is to remove the now redundant connections g and h in the above figure and to finally go into the list of components and remove the ideal components zone_a and zone_b. The result will be a network of 10 components, 11 connections and 10 containments.

```
Connections

   Sending comp  @   Node        to   Receiving comp @   Node     | Conn Type | Mass Div
a outside air     ambient        -->  inlet_duct      air node 1    zone/amb    1.000
b inlet_duct      air node   1   -->  supply_fan      air node 1    to compt    1.000
c supply_fan      air node   1   -->  heater          air node 1    to compt    1.000
d heater          air node   1   -->  supply_duct     air node 1    to compt    1.000
e zone_a          zone air       -->  duct_ret_a      air node 1    zone/amb    0.500
f zone_b          zone air       -->  duct_ret_b      air node 1    zone/amb    0.500
g zone_a          air node   2   -->  duct_ret_a      air node 1    to compt    1.000
h zone_b          air node   2   -->  duct_ret_b      air node 1    to compt    1.000
i duct_ret_a      air node   1   -->  mixing_box      air node 1    to compt    1.000
j duct_ret_b      air node   1   -->  mixing_box      air node 1    to compt    1.000
k mixing_box      air node   1   -->  duct_mix_fan    air node 1    to compt    1.000
l duct_mix_fan    air node   1   -->  exh_fan         air node 1    to compt    1.000
m exh_fan         air node   1   -->  exh_duct        air node 1    to compt    1.000
  ----------------
+ add/delete/copy
```

Figure 13.15 Connections after editing.

13.4 Links to zones and controls

At the bottom of the network definition menu there is an option link plant to zone. Before we can use this facility we need to define two zone controls and that requires saving the network of components and changing to the zone controls menu to initialise the controls and then return to the network component interface to complete the process. Perhaps a future version of ESP-r will include a wizard to sort this out...

The first zone control senses the temperature in zone_a and actuates at the air node of zone_a and has one day type and one period in that day and the control type to flux connection between zone and plant but skip filling in the details.

The second zone control should sense the air temperature in zone_b and actuate at the air node of zone_b and have one day type and one period with the flux connection control law (see Figure 13.16). While you are in the control facility there is an option to set (another type of) linkage between the control law loops you just created and the relevant thermal zone. When you have done this save the zone controls.

Now return to the network of components and select the item link plant to zone. The control file will have been scanned and there should be two entries, the first for connected zone zone_a and the second for connected zone_b, both with a convection type connection. The remaining fields define the nature of the supply and whether there is an extract.

The link for zone_a uses the component supply_duct as the supply and the duct_ret_a component is the extract. The link for zone_b uses the component supply_duct as its supply and the duct_ret_b as the extract.

The interface will look like Figure 13.17 when this is completed. This is a good time to save the network of components. You will be asked whether the zone controls should be updated to reflect the recent changes in the

network of components (say yes).

Almost finished. The zone controls need to be adjusted. The linkage function guessed at the capacity of the heater component and will have set the heating and cooling capacity of the zone control to a value which is incorrect (the actual value of the heater 3000W and 0W cooling).

Up to this point we have used zone controls to establish the link between the thermal zone and network component domains and some of the parameters in the zone controls (e.g. the capacity) are based on information used during the definition of the network of components.

We now have to define the logic which will drive the heater component and for this we must define a so-called plant control. There will be one control loop, it will sense node one within the duct_ret_a component and it will actuate node one within the heater component. The controller type is senses dry bulb actuates flux (from within the list shown in Figure 13.12). There is one day type and three periods during the day. From 0h00 to 7h00 the control will use a period switch off control, from 7h00 there will be an on-off control with a heating capacity of 3000W and a cooling capacity of 0W and from 18h00 a switch off control.

The selection of control laws (see Figure 13.18) is somewhat terse, but the help message clarifies the relationship between the control law and the control type. These relationships are required because some components work on flux and some on flow and the actuation needs to reflect this.

A word about the data for the on-off control period. There are seven parameters:

- mode of operation (1.00)
- off setpoint (23C)
- on setpoint (19C)
- output at high (3000W)
- output at low (0W)
- sensor lag (zero) actuator lag (zero)

When the plant control is complete and saved it is a good idea to generate a fresh QA report for the model. This will provide additional feedback for checking that your model is consistent.

After you have reviewed the QA report adapt the simulation parameter sets. Use a 15 minute timestep for the zone solution with the plant simulation at 10 timesteps per building timestep and ensure that there are names for the zone and plant results files filled in.

Commission an interactive simulation. If all went well the simulation will take a few minutes to run (the plant is solving every minute). When you go to look at the performance prediction look for performance graphs such as in Figure 13.19. The upper olive-gree lines are the heater temperature and flux output (labeled as other). The lines below are the temperatures at various points in the ducts. It is also worth looking at the performance characteristics reports for the zones.

```
Controls

a control focus >> zones
b description: no overall control descript
c description: no zone control description
  loops      :  2
d link loops to zones

 ----------------------------------------
 cntl| sensor |actuator|day |valid |period
 loop|location|location|type|during|in day
e   1  1  0  0  1  0  0   1   1 365   1
f   2  2  0  0  2  0  0   1   1 365   1

  ----------------------------------------
+ add/delete/copy control loop or day type
! list or check current control data
> save control data
? help
- exit this menu
```

Figure 13.16 Initial zone control settings.

```
Linkages

   connected  |  connection |  connected     |  connected     |
     zone     |     type    |emmiter/supply  |   extract      |
a zone_b       convective    supply_duct      duct_ret_a
b zone_b       convective    supply_duct      duct_ret_b

  ----------------
+ Add/Delete/Copy
```

Figure 13.17 Completed linkages between zones and components.

One of the reasons one might need to use a network of components is to inquire into the internal state of the components so this is a good time to review what is on offer and, importantly, what information about the components are required to recover performance data.

One way of discovering the performance sensitivity of networks of components to changes in control parameters is to run a series of simulations typically changing one aspect of a control or a component parameter at a time. This process works even better if a friendly control engineer takes part in the exploration. Certainly an on-off control will result in different performance characteristics to a PID controller, but remember to walk before you run when it comes to PID controllers!.

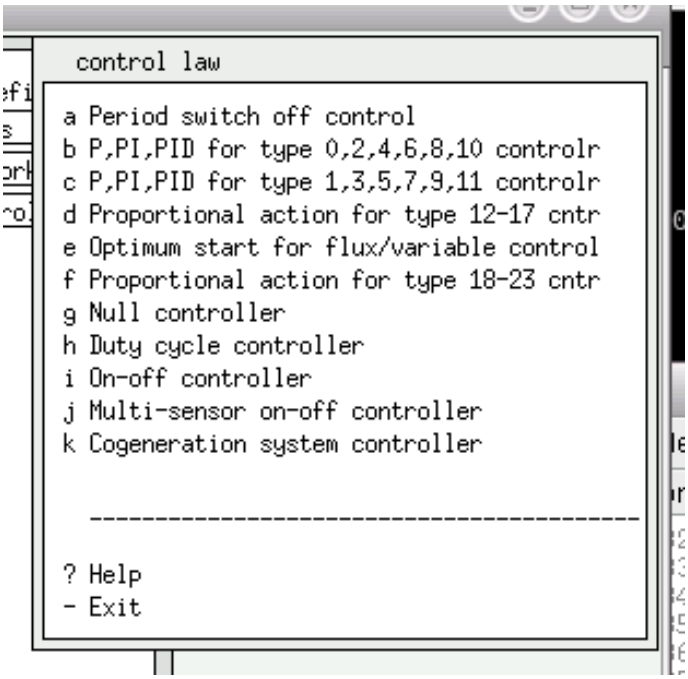


Figure 13.18 Component control laws.

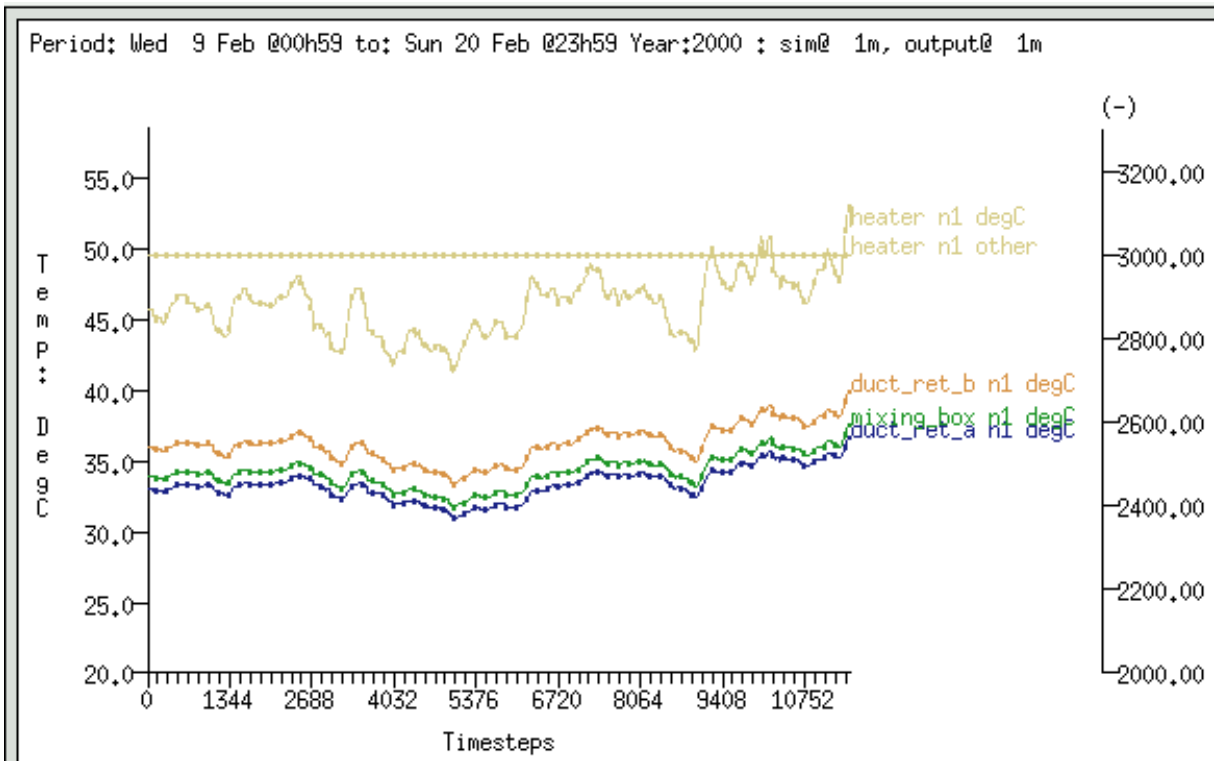


Figure 13.19 Performance predictions within model with thermal zones.

Judging the additional resources needed in comparison with ideal zone controls is best done once you have defined a couple of component networks and developed some proficiency at the tasks involved. The goal is to employ the most appropriate approach to a given simulation project and only using complex facilities where a less complex approach does not support the requirements of the project.

14 Working procedures

Simulation teams who are attempting to deal with real designs in real time can deliver less than they intended or work harder than they intended for a number of reasons:

- missing out a brief, but critical step in a sequence of tasks
- failure to review a model for inconsistencies as it evolves
- failure to record critical assumptions or failure to find notes about critical assumptions
- misunderstanding the nature of the assessments to be carried out
- hacking model files for a quick change without documenting the change, checking that the change was syntactically correct or running calibration tests
- assuming that everyone else knows that *ext_glas* is intended for use on the *only* the south facade of the building

Such self-inflicted errors and omissions are only the tip of the iceberg. Software vendors, even if they wanted to, have little or no influence on the organisation and procedures of simulation work groups let alone the form and content of interactions between team members.

The following discussion is an expansion of ideas that the author and others contributed to *Building energy and environmental modelling: CIBSE Applications Manual AM11: 1998* The Chartered Institution of Building Services Engineers, London, April 1998. And, as is mentioned in the *Model Quality* chapter, the recommendations in AM11 continue to be valid and should be on the bookshelves of every simulation group. From the perspective of the *Cookbook* AM11 tends to be understated in its recommendations.

The *Cookbook's* definition of *QA & QC* extends the traditional risk management and consistency checking found in AM11 to include procedures and organisational patterns that help identify opportunities for delivering additional value to the design process. Many of the extensions to standard practice require minimal resources. They are not rocket science although they require attention and flexibility from members of the simulation team as well as others in the design process.

Simulation teams who thrive invest considerable passion in working practices which compensate for the limitations in their tools. This chapter is, in part based on a decade of observation of successful and un-successful simulation groups. Space constraints have limited the number of examples of working procedures and the depth of discussion of their details. Consider this chapter as a starting point for

creating your own working procedures. The complement of this chapter is the *Model Quality* chapter which expands some of the issues raised in the following text.

14.1 How can the vendor help?

In one sense, the topics covered in this chapter are independent of software vendors because there are many possible ways to deploy simulation tools and as many preferences of use as their are simulation teams. Although specific examples are given from the context of ESP-r, users of other software will probably find much that is familiar although some adaptations may be required on the working procedures to fit the conventions and facilities of another tool.

Vendors do have influence - their marketing to managers is *spun* to emphasise productivity gains and ease of use and their brochures are full of astoundingly complex models. Once the license is purchased vendors have more incentive to train users on keyboard skills than to support managers who wish to evolve better working practices. There is also little incentive to help managers understand what constitutes a model which is well-suited to the needs of a particular design project.

Vendors mount training courses which are optimised for keyboard skills, production tasks and rapid generation of models. Less attention is paid to the more esoteric skills (e.g. model abstraction, interpretation of results). Materials which address these issues may be available if you know who the ask and what to ask for.

Vendors influence is greatest in the core design of the tool. There are two general classes of shortcomings which impact simulation teams. The first the the tendency to equate model quality with *if what you see looks correct it must be correct*. Optical illusions abound in simulation so simulation teams must employ robust procedures to enforce model quality.

The second shortcoming is that it is far to easy for independent actions by multiple users to cause clashes which are difficult to resolve. Although simulation data models are among the most complex dreamt up by humans they are not implemented via database management tools that support simultaneous user interactions. This limits the deployment of simulation in distributed work groups.

14.2 Responsibilities within simulation teams

Although there are successful independent simulationists, simulation tends to be most powerfully deployed in a team environment.

Why would this be the case? Firstly, simulation demands a range of skill sets e.g. managerial, technical and communication skills.

Second, projects can fall short of their goals if errors in methods or descriptive details are not identified. Some individuals can perform the mental shifts required, but self-administered QA is an invitation to risk that is difficult to overcome.

Third, as simulation tools become more complex and their facilities expand into new analysis domains, it is

increasingly difficult for an individual to be proficient with all aspects of a tool or to manage the volume of detail included in complex models. Real designs often imply a level of complexity which can only be managed within a team environment.

There are, of course, options for individuals to work jointly in ad-hoc teams using the facilities and techniques of simulation teams who are geographically distributed.

For most projects, the limitations in sharing models and maintaining a virtual team is not the speed of the internet, but the inability of simulation software to cope with asynchronous manipulation of models and, to a lesser degree limitations in the underlying data model which make it difficult to embed documentation and assumptions required by well formed working practices or maintain a log of actions taken.

In an ad-hoc or formal simulation team there are a number of participants:

- The team manager, who works from the perspective of project goals, client requests, resource limits, delivery dates and staff motivation
- The quality manager, who helps with calibrating the model, ensures the model is fit for purpose, that predictions are as expected and who (ideally) is looking for opportunities to add value to the deliverables
- Simulation staff, who implement the work plan, coerce the tool to fits the needs of the project, commission assessments and carry out production related data extraction and interpretation of predictions
- Domain experts who, on an ad-hoc basis, help to define the modelling approach, confirm predictions and identify opportunities and glitches
- Mentors are also ad-hoc participants in the process during the planning process and tend to be focused on creating new working practices and augmenting staff skills.

Setting up a simulation team is not a trivial task as can be seen in section 4.2.1 of CIBSE AM11. Time has not diminished the critical truths in the section on human resource requirements.

14.3 Classic mistakes

Observations over a number of years indicate that simulation teams tend to underestimate the time investment required to create and evolve working procedures. Similar optimism pervades the task of maintaining and extending staff skills.

Conference papers tend not to hi-light the risks associated with team managers who hold themselves aloof from knowledge of the tools used and are thus unable to adapt their directives to take advantage of the strengths of their staff and tools. Bluntly stated - errors and omissions in management are no less important than errors and omissions in technical execution.

There are any number of working patterns that result in a misallocation or misuse of the skills of staff. Each of these sub-optimal working patterns has one or more alternatives which make

better use of tools and/or staff skills.

First there is the classic confusion that keyboard proficiency implies domain skills. Opinionated users with appropriate background skills to guide their use of the tool have a distinct advantage what ever their speed of interaction.

The next classic mistake is to rely on raw computing power rather than well formed working practices and well designed models. Yes, there are cases where computer power limits productivity, and the majority of simulation tasks are constrained by other factors.

Managers often believe that *user friendly* software allows junior staff to function with limited supervision and/or limited training. What tends to happen is that junior staff lack the self-restraint to avoid complexity and become driven by the tool.

A perfect storm for a simulation team is a manager who gives an inappropriate directive to a novice who either does not have the background to recognise the directive is suspect or the confidence to request clarification. The novice thus works very hard at digging a hole for the team to fall into and lack of attention ensures the pain is distributed.

Another classic tempting-of-fate is an assumption that ability to *generate* reports and graphs equates to a an *understanding* of the patterns within reports and graphs sufficient to add value to the project. Valuable patterns within a data set can remain hidden just as indicators of error can pass unnoticed.

Such paths to failure can be avoided. For example, those with the maturity and intuition needed to quickly and accurately recognise patterns in performance predictions are often found in a nearby office. That such valuable assets are out-of-the-loop is a self-inflicted limitation. Bring them into the conversation.

Just because the tool interface includes a WOW feature, you still need a good reason to select it. Auto-generation or auto-sizing should not be invoked on a casual basis until the team reviews such facilities. Is the in-built facility based on rules and concepts that the team agrees with? Are the resulting entities understandable and do they conform to the requirements of the project?

14.4 Planning simulation projects

The author once had a consulting project to evaluate whether mechanical dampers could be used in the facade of an office building to provide natural ventilation during transition seasons. The model included eight zones, internal mass, shading devices, two variants of air flow network and three control schemes. A synopsis of the findings was available five hours after the plans and sections were first opened.

This was possible because the first hour and a half was devoted to planning the model:

- establishing what needed to be included in the model
- defining the extent of the model and its resolution
- establishing critical co-ordinates in plan and section
- planning the zoning of the model
- review available construction databases and identify additional elements needed
- planning the sequence of tasks that would limit error and allow for entities to be re-used
- planning the calibration tests to be carried out
- sketching out the model, deciding a naming strategy
- sketching out the air flow network and gathering relevant data
- reviewing an exemplar model which used the same control logic

With this information the task of creating the initial cellular office and facade elements was straightforward and attribution proceeded without interruptions. It was then possible to use these elements in many other parts of the model. There was no need to use a calculator or pause during the input process because all of the critical dimensions and attributes were available for checking.

About one and a half hours was spent creating the model and a half hour was spent in calibration. The remaining time was *living with the model* and testing out different damper controls and drafting the synopsis.

Spending one third of project resources for planning and data gathering is, for many experts, a typical approach. If less than a quarter of the project is used for preparation then the risk of delays and errors later in the process increases.

The following table includes a number of issues which may confront simulation teams in the planning stage.

Core issue	Related issue	Actions
Does the client/design team have prior experience with simulation based assessments?	Prior experience eases the *educational task* otherwise client expectations could be an issue.	Ensure time and resources for clear communication with the client. Brief the client on the nature of the information they will be asked to provide and consider their response.
Does the client/design team know the types of performance data which can be generated and the available reporting formats?	Management of reporting expectations. Identification of how interim results can be communicated.	Review client preferences as well as clarify potential misunderstandings in deliverables.
Has the client used language or provided sketches that indicate *beliefs* about how design will work? Simulation can test such beliefs.	The best time to test beliefs is as soon as they are noticed. The creation of quick focused models which can deliver information with minimal delay is critical.	Confirm that staff have the skills to deploy the types of focused models needed in this project. Confirm if typical data for the type of building if available. Create a test case to confirm that staff can extrapolate from sketches and capture the essential characteristics of a design.
Has the client indicated what criteria would signal success?	What magnitude/frequency of change in that criteria? What additional performance data might be captured to help the simulation team calibrate the model?	Confirm if criteria in keeping with current practice or is it a client-specific definition. Investigate whether the criteria provide useful indicators for what-if questions that the client has not yet raised.

Core issue	Related issue	Actions
Has the client indicated what criteria would signal failure?	What magnitude/frequency of change? What are likely modes of failure? What needs to be measured to identify risk?	Confirm criteria are in keeping with current practice or are client-specific. Evaluate likely assessments that will test how robust the design is. Identify operational regimes or boundary conditions that would help identify risks.
Is the design team searching for improvement on a range of issues or a single issue?	Multiple issues might require several models or model variants.	Confirm if staff can cope with multiple models and/or model variants likely in this project. Clarify how are different models or variants to be identified.
Are what-if questions in the form *what happens if we use a better quality low-e glass* or *what happens if we use product X*?	The role of the simulation team in the design process may need to be flexible. Is it to provide information for other to make decisions or is it pro-active?	Identify methods might to identify *a better quality low-e glass*. Raise the issue of wither a supplied specification is appropriate and in the clients interest.
Do what-if questions pose parametric questions (e.g. which skylight area between 12 and 36% is the point where cooling demand escalates)?	The distributed file system used in ESP-r models has the potential to simplify parametric studies or complicate them depending on the nature of the parameter to be changed. Early confirmation is essential.	Confirm tool supports the creation of model variants required for this project. Discuss whether it will be necessary to generate scripts to automate parametric tasks. Confirm the script is correct and that correct data has been extracted. Consider if scripts and methods re-usable.

Core issue	Related issue	Actions
Is this project similar to previous projects? Can we adapt a past model for use in this new project?	What did we learn in the previous project which could be applied? What were the difficult issues in past projects of this type?	Confirm if staff have the skills to re-use and adapt existing models. Confirm if the existing model well documented. Confirm log of the procedures and staff resources used to see if the current resource allocation is appropriate.
Does the current state of the model reflect the ideas and concepts developed during the planning stages?	Is the complexity of the model consistent with the resources allocated to the project? Have the resources used matched the initial plan?	Consider if staff tasks need to be adjusted. Consider if additional staff required. Plan for contingencies if staff become ill.
A potential project will be discussed with a client in a meeting tomorrow.	Who should take part in the meeting? What is the near-term work load? What key phrases are important to listen for? How much do we need this project?	Review details of similar projects and the staff involved. Discussion criteria for deciding whether to bid on the job. Ensure that presentations and sample reports are available if the client needs to be briefed on team capabilities.

14.5 Team manager

It is critical that someone considers the broad sweep of issues within the project as well as the clients goals. Few can manage the mental leaps needed to shift from the detailed focus of simulation use to a broad perspective. It is *usually* considered good practice to ensure that simulation staff have access to someone who is paying attention to these *other perspectives*.

It is possible to adopt pro-active or passive approaches to team management as long as the frequency of communication and the rigourous testing of the plan and the deliverables is maintained.

The team manager may be appointed on project basis or hold the position for

a number of projects. In some teams individuals service several positions - in one project they may act as the team manager, in another they may carry out simulation tasks and for another they focus on quality issues.

Flexibility has several benefits:

- It allows ad-hoc teams to be formed with the best available talent.
- It ensures that staff who have a broader range of skills and have opportunities to exercise and improve those skills.
- It can distribute skills to recognise what simulation tasks are straight forward and which are likely to be tedious.

It is particularly helpful if a simulation team has access to those with experience in the operational characteristics of buildings as well as in strategies for solving design problems. Such knowledge, if deployed interactively, can speed up the process of testing design issues.

Remember, others in the design process may know very little about the work of simulation teams and what information they can deliver. If the simulation team has sufficient confidence to allow others into *the process*, there are significant opportunities for both parties to discover concepts and ideas that will be of mutual benefit.

And one of the benefits of interactive working is that it gets around the rigid structure of formal reports and the limited topics that can be included in reports. Fifteen minutes of browsing performance predictions (the ESP-r results analysis tool is specifically designed for interactive use) may uncover patterns that would not have been included in a report might have taken hours to format for a formal report.

Staff rotation and interactive working with others in the design process also prevents the isolation and dead-end horror stories that many simulation staff experience in some companies.

Another type of flexibility is in the selection of the most appropriate tool for the task. The *Cookbook* is written in the context of ESP-r. Simulation teams may have a preference for one simulation tool and if you are reading the *Cookbook* then ESP-r may be your preference. It may be possible to coerce your preferred tool to carry out a range of tasks but the lack of choice may impose a cost. There are virtual physical descriptions that other simulation environments provide which might be more appropriate for a specific design.

The *Cookbook* recommends that simulation teams review the capabilities and cost-of-use of simulation tools and develop selection criteria. The costs associated with acquiring and supporting multiple tools should take into account whether staff are able to cope with multiple tools or if additional staff are required.

Project planning should consider whether there will be a need for a mentor or one or more domain experts. Early warning of a possible request can reduce the risk of such people not being available. They may provide useful feedback to the project. Consider - if an expert is needed to help with a

crises, would if have been less costly to have retained them at the start of the project and perhaps have avoided the crisis?

14.6 The quality manager

Just as the author of a book needs an editor to help complete the story, neither the team manager nor simulation staff are in a good position to recognise whether the model continues to be fit for purpose, whether the latest performance predictions are providing a consistent story or include a new pattern which needs attention.

Good pattern matching skills and eye for detail are core competencies. A traditional approach is for quality managers review reports generated by others and request clarification from staff.

And *what if* the quality manager did not have to wait for others to generate reports and did not have to look over their shoulders to see the current state of the model and/or assessments? What must phrases such *come over to my screen and see what I found in the Monday morning start-up test period* originate with simulation staff?

An ability to use simulation tool facilities to review models, generate reports about the models breaks a classic dependency. An additional pair of eyes with pattern matching skills *and* different goals increases the probability of discovering opportunities as well as faults. A decision to invest in such skills for quality managers transforms their mode of working from passive to active.

Such pro-active investigations need not be a burden in time or computing resources - if the group has methods in place to target simulations on specific short period assessments. Indeed, the criteria one might use for initial calibration runs would often do dual-duty as periods for the quality manager to use. In the case of ESP-r, it is possible to pre-define simulations and embed this information within the model.

For many simulation tools the skills necessary for browsing a model and invoking a pre-defined simulation can be acquired in a few hours. Of course, more time is needed to become comfortable with browsing performance predictions and generating graphs and reports.

Productive quality managers will have evolved strategies for reviewing models as well as reviewing simulation predictions. They will have strategies to ensure that they can track the activities of others or track the evolution of models so as to identify points of interaction.

Just as in financial markets there is a *moral hazard* in expecting the quality manager to catch and solve all glitches. The *quality process* only works if others in the design team take steps to reduce risk as models are initially defined and evolved.

Other chapters in the *Cookbook* were written from the perspective of reducing errors as creating models are easy for others to understand. Quality managers appreciate models that tell a clear story.

One task of the quality manager is to be a champion for reserving project resources for the exploration of value added issues. This requires tracking the

progress of the project and ensuring, where possible, that there is time available for exploring issues that the client has not yet raised.

Such resources may be in the form of time for simulation staff to undertake speculative explorations or it may relate to a current or past project.

As the *Cookbook* has mentioned in other sections, efficiency gains should be directed at freeing up time to *live with the model and explore its performance* and thus better understand how it works.

Typically the quality manager will have a list of classic questions and issues to explore within the model.

- ☐ If there is an early peak Monday demand is it possible to use an optimal start regime?
- ☐ Is there excess heat stored in the fabric of the building overnight? Check if a night ventilation purge or extending the running hours of the environmental control system will correct this.
- ☐ Are conference rooms subject to rapid overheating if the room has full occupancy? Test is massive partitions dampen temperature swings and improve comfort?
- ☐ Does the environmental controls short-cycle. Check if reduced capacity will correct this.
- ☐ Is the building over-capacity in terms of heating and cooling? Check how many additional hours over the set point occur during a season with a 5% reduction and compare this with the reduced capital and running costs.

The table that follows illustrate some of the issues that confront quality managers.

Core issue	Related issue	Actions
Is the current project similar to a past project?	Is there valuable information in the project notes and documentation? Do staff remember useful/critical issues from that project?	Discuss findings with the team manager as well as the simulation staff. Review the past models with the team and identify critical issues to be tracked in the current project.
Are there design issues in this project which are new to the simulation team?	What might be valid approaches for each issue? Can a simple model allow the team to explore this issue prior to implementing it in full scale? Should the mentor be called in?	Review criteria used to identify which approach is best. Find out who needs to work together to test out the approach. Confirm points in the work flow where interactions with the quality manager are needed.
What naming scheme will clarify the model to others in the design team? Is the client visually oriented or number oriented?	Does the client talk about the project using phrases that could be embedded in the model? Are the essential differences in model variants communicated by the naming scheme?	Confirm if the simulation tool cope with the naming scheme? Consider whether manufacturer's names for products appropriate for use in the project. Confirm which performance data can be presented in visual form to match client preferences.
What documentation has the client provided?	Has a review of this documentation identified questions that need to be resolved prior to work starting?	Confirm if some or all of the client documentation be held with the model. Advise on who should embed this documentation.

Core issue	Related issue	Actions
Does the client require a record of the tasks undertaken, the method(s) used and the model?	Are staff keeping a log of what they did, what assumptions they made and what information sources they used?	Show the model to a third party and see if they understood the essential attributes of the model.
Does the model continue to reflect the initial plan or does the plan need to change? Are there resources available for value added opportunities to be explored?	Is the actual complexity of the model in the same order of magnitude as the initial plan?	Investigate if delays in tasks could make it difficult to keep to the plan. Ensure updates to the model been broadcast to all interested parties. Outline value added issues that need to be discussed.
The model predictions indicate 15% less heating capacity - is this an error or is this an opportunity?	What changed in the model? Who made the change to the model? Who can corroborate the difference?	Identify other performance data which could be influenced by this change. If an opportunity, investigate further changes change to the model to further improve performance.
During checking it was found that occupancy in several rooms was less than specified.	When did this happen? What performance data could be at risk? What design decisions might be at risk?	Check wither the planned occupancy density or the model details are correct. Review with simulation staff how predictions change when the occupancy is updated. Broadcast the changes and whether change is in line with project goals.

Core issue	Related issue	Actions
The project is slightly ahead of schedule what do we do now?	Where are opportunities for making the design work better? What is likely to be the next issue that the client will ask us to consider?	Revisit the assessments looking for likely performance improvements. Investigate modifications needed to explore likely topics. Employ a focused model to check if an alternative approach could provide better performance data.

14.7 Simulation staff

Traditional deployment of staff involves junior staff *primarily working* to create models, run assessments and extract performance data. The critical issues for successful deployment is the level of experience junior staff have in addition to their tool-related skills and their status in the group.

Staff who lack experience in a particular design or engineering domain are at a disadvantage in comparison with more opinionated staff. Although tool vendors do not advertise it, those with strong opinions are much more likely to make good use of simulation tools - opinions help in the selection and use of tool facilities. Managers who wish to overcome this limitation might consider pairing the novice with more experience staff for several projects.

Those with opinions will certainly have much more to contribute to the planning stages of a simulation-based project in terms of:

- Information required from the client at different phases of the work
- Feedback on likely simulation tools for the project e.g. benefits and drawbacks of different tools, time and computational resources implied for different tools.
- Ideas about how design issues might be represented within the simulation environment(s)
- Ideas about zoning strategies and level of detail implied for various design questions
- Ideas about past models which may be useful to review

There is a place for novices in a well-formed simulation team if they have access to more experienced staff as well as the mentor and there is frequent and close supervision.

Where simulation staff are included in discussions, are able to buy-into the goals of the project and have sufficient status to creatively evaluate directives and suggest alternatives, there is little chance for a perfect storm to develop. Take one or more of these away and risk increases.

The following table includes typical issues which confront simulation staff.

Core issue	Related issue	Actions
What preparation is needed for this project?	Are we clear about the client's ideas that the model should confirm? Have we done similar work in the past? Do we have test models for this building type?	Meet with team manager to review client requirements. Sketch likely approaches to the model and review with the team Review past projects and identify if there is an option for adapting prior models. Review library of test models for candidates for use in testing ideas and techniques for the current project
What analysis facilities are required to support the project?	What running cost/control issues? What comfort/air quality issues? Are thermal bridges an issue?	For each domain identify what needs to be measured. Define the level of detail required. Identify likely interactions between domains.
What model calibration approaches are appropriate	Is there a range of best-practice performance indices? Can a previous project be used as a benchmark? What operational and climate conditions would be a good test of the response of the building?	Review standard literature and past reports. Consult working procedures for which types of tests are recommended for this building type. Identify climate pattern(s) and operational characteristics to use in tests. Embed appropriate simulation directives in the model so that calibration runs can be undertaken as the model evolves.

Core issue	Related issue	Actions
How much time will it take to create the model?	Is there a record of resources from similar projects? Is this a crank-the-handle or exploratory model? What staff would work well in this project? What staff productivity can be assumed?	Review past projects of this type and review with team manager. If an exploratory model plan a series of proof-of-concept models as a benchmark. Review with mentor. Review tasks required vis-a-vis available staff and staff who have worked on similar projects. Ask each staff member to provide an estimate of time for preparation and model creation and review with the team manager.
What model variants might be needed?	What other questions might the client pose? Is the current model at the edge of the tool facilities or staff skills? Are changes directed by new information provided or are they general parametric variations?	Discuss with mentor and domain experts what other topics might be addressed with variants of the current model. Review working practices for resource estimates needed to increase or decrease the resolution of the model. Review current model complexity in each domain and discuss this with mentor and project manager. Locate a test model and use it to test automation facilities or scripts. Report findings to team.

Core issue	Related issue	Actions
What zoning pattern should be used in the model?	What is the distribution of occupancy patterns? What portions of the building are sensitive to the facade? What variants of control logic might be used in different sections of the building? Is stratification an issue? Is cross-ventilation or air flow between rooms likely?	Using separate overlays sketch out regions for occupancy (density, schedules), control (logic, schedules), perimeter sensitivity, air flow connections, system types and control-ability. Unify the sketch overlays for initial ideas. Work out an alternative sketch of zones taking into account likely future issues. Sketch scenarios for air flow networks and revise zoning to accommodate.

14.8 The mentor

Different groups have different definitions of mentor:

- A person who has mastered the simulation software and helps staff to become comfortable with features which tend to be viewed as either *magic* or where *dragons live*.
- A person who has undertaken simulation projects of the scale, complexity and mix of domains which are being considered by the simulation group (to provide guidance).
- A person who knows the physics of a design issue and who works with a team to help define strategies and evaluation criteria (e.g. a domain expert).
- A person retained to carry out extended training within the simulation group, typically to assist the team to enter a new market or work with different types of clients.

Mentors may be part of the team or may be outside consultants who are retained by the team via a support agreement (X hours over the next 3 months). Skill sets may be focused (e.g. they know how to design and deploy air flow networks for large spaces) or broad (e.g. they have managed and/or delivered similar projects).

Sometimes simulation teams engage a mentor because they want to explore new topics or explore an alternative way of working within an active project (and want to limit their risks). In such cases mentors usually are given the authority to take over tasks being carried out by staff if that is what is required to guarantee deliverables.

In other cases a mentor may be on-call for brief consultations because their

experience allows them to quickly answer questions or demonstrate a technique. This could be done in person or via a video conference.

One rapid approach to evaluating simulation methods (and tools) is to identify a recent project and explore one or two issues in the project via the use of simulation. Participants would then compare what simulation delivers with the prior findings in terms of resources required, skills needed as well as a comparison of predictions.

Recent projects are especially good if staff remember the approach they took and have access to the underlying project data. The mentor can both guide the simulation team and help them to understand the predictions from simulation.

The following table includes issues from the point of view of the simulation team and from the mentor's point of view.

Core issue	Related issue	Actions
Team: The project starting next week involves X. We have limited experience - how shall we proceed?	Is this a topic that the mentor deals with? Who needs to be involved in the discussion? Do we upgrade or use our standard approach?	Specify the issues, points of confusion, time frame and goals. Identify staff to work with the mentor and clarify their goals. Define *upgrade or discard* policy.
Team: Fred is not satisfied with how he works with facility X.	Is this in the interest of the group to improve this skill? Does the mentor handle this issue? Is this part of a larger issue to be dealt with by a mentor or via a standard training course from the vendor?	Check if this caused delays or errors. Check if new skills will free up time for other activities or help other team members. Get time/resource estimate from mentor and check for time scale and cost of the vendor's course.
Team: time estimates were out by 30% in the last project. Is there a different approach that would work better?	What does the mentor need to understand about the group in order to evaluate that project? What methods could be use to test alternative approaches?	Create a synopsis of the project for review. Review synopsis with design team and get initial feedback. Re-enact the project to identify faults Identify staff who should take part. Agree a timetable for review, re-enactment and work sessions.

14.9 The domain expert

Domain experts can play different roles within a simulation team. For example, their domain may be the theory of computational fluid dynamics or experience with wind tunnel testing and how to integrate experimental data into the virtual physics.

The domain expert differs from the mentor in that the goal is to solve or understand a project based issue rather than adapt the teams working practices or improve staff skills. The domain expert may know little about the

specifics of the simulation tool but will likely have strong opinions about they see in performance prediction graphs or reports.

The domain expert can supply a disinterested *second opinion* for a contentious issue within a project or help verify that current predictions are in line with expectations.

Remember that the issue for the expert is likely to be both **normal and obvious** to them. They may not realise that others do not share this knowledge. This can cause confusion to both parties. Take steps to reduce this confusion if at all possible.

The domain expert may have *opinions* that includes the *fact* that their approach is the only possible way to deal with a specific issue. This may or may not be true. If there actually are several valid approaches then the expert's contribution may be to provide information their approach and then the team must evaluate this against the other options.

If there is only one valid approach and the simulation tool does not support that then there are several options:

- check to see if the simulation tool can be adapted
- check to see if there is another tool which uses this approach
- evaluate whether there is a way to coerce the tool to approximate the approach suggested by the expert
- if it is early enough in the project check if this part of the analysis can be subcontracted
- if the project is dependant on this issue being resolved and it cannot be resolved within the time or resource limits of the project consider withdrawing

The following table includes several scenarios related to experts:

Core issue	Related issue	Actions
The expert and the team use different terminology.	Are both parties talking about the same issue? Is the terminology different because there is a difference in the underlying approach used by the expert and by the simulation tool?	Show the expert a list of common definitions of the jargon used within the group and within the simulation tool. Arrange for a meeting to discuss underlying methods and terminology.
The expert asks to know how the simulation tool treats a specific issue.	Is the information available? Is it in a form which can be understood? Is there an example model which demonstrates this issue?	Scan the source code for relevant blocks of code or pass the question to the software vendor. Scan the documentation for relevant sections and/or ask the software vendor if there is additional documentation. Set up a session to explore relevant models to confirm if a) the expert understands, b) the expert agrees with the facility within the software.

14.10 Staff productivity

Given the same simulation tool, the same computer type and the same brief two different simulation staff can use radically different times to arrive at a completed model. What a novice can produce in a frustrating two days, seasoned staff can produce in ˜two hours with a computer which is half as fast.

That difference in productivity is expected and should be factored into the staffing of a project. A tight time-schedule may be best supported by using more experienced staff whereas those with less experience will be more comfortable with a project with less challenging deadlines.

When two staff with roughly the same capabilities take radically different times for the same work then the manager and mentor should investigate. Chances are the biggest difference will be the strategies employed. Strategies can be learned.

If possible, the quality manager should devise task sequences to confirm the skills level, strategies and *inventive resources* of staff.

Simulation staff who are working at the limit of their skills are less efficient and more likely to generate errors than staff who are well within their competence level. Therefore, a well formed working procedure *will ensure* that staff are working at a pace that is sustainable. Simulation staff should be clear about their current productivity and inform management if they are at risk.

It is a challenge to match resources *actually used* with assumptions made in initial planning. One major contribution of simulation staff to the planning process is to provide realistic estimates of time and computational resources. Keeping notes about the time actually taken for specific tasks should be included in working procedures. Frequent updates to estimates as real data becomes available can help in the management of staff resources.

What about time estimates for new (to the group) tasks? One could guess and risk under or over bidding. Or the simulation team can devote some of its resources to anticipating new topics and tasks. It can compile background documents, create exploratory models and undertake mock projects in preparation.

Simulation staff should be pro-active and request time (say 2-3 hours a week) for speculative explorations and keep the mentor and the team leader informed of progress. It is just as valuable to report a tool facility that is not ready for use in consulting projects as it is to discover a virtual measurement which can be included in future reports.

14.11 Summary

A well formed working procedure will ensure that staff are working at a pace that does not exhaust their mental reserves. This suggests that all members of the team should be clear about their own level of competence and how much reserve capacity they have. And they should not keep this a secret and there need to be communication channels for such issues.

It is a challenge to match resources used for generating and testing models with assumptions used in initial planning. With experience, simulation staff, mentors and quality managers can give close estimates of the amount of time they expect to take.

Well formed working procedures will ensure that before a bid is given to the client there is an evaluation as to the fitness of the suite of software tools available vis-a-vis the likely demands of the project. There will also be an evaluation of the current skills level in case preparatory work is required.

15 Model Quality

Simulation teams who are attempting to deal with real designs in real time are confronted by the need to ensure that their models are syntactically correct as well as semantically appropriate for the project.

Simulation teams who thrive invest considerable passion in ensuring the quality of their models. They do so to limit risk (a traditional reason for QA and QC). The *Cookbook* also advocates working practices that ensure model quality because of the early warning of opportunities to deliver additional value to clients.

Model quality involves both the facilities of the simulation tool and the skills of the simulation team. Among the most important issues are:

- designing models that the simulation team can understand
- designing models that clients recognise
- identifying aspects of models that tool checks may not recognise as correct
- identifying aspects of models that the team may not recognise as correct
- working practices that bypass inbuilt tool checks
- working practices that bypass the quality manager
- frequency of model and calibration checks
- understanding model contents reports

Semantic checking is related to the *design of the model* - how it includes and excludes thermophysical aspects of the design. Because it is an art as much as a science it is more difficult to establish rule sets for model design. Here the issues are:

- models (or tools) that are *not quite fit* for purpose
- models which continue to be used after entropy has set in
- models which are stuck in an unusable state

15.1 How can the vendor help?

The quality of models **begins** with the facilities offered by the software vendor e.g. QA reports, documentation, training and in-built software checks. Vendors decisions about in-built facilities have a substantial impact on the resources that simulation teams invest in model checking.

Some vendors believe their marketing departments spin that *what you see is what you get*. Unfortunately, what you see on screen is only one of many possible views of the contents of a model. Interfaces can present users with

optical illusions. Here are some examples:

- A wireframe view can look correct but have reversed surfaces or missing surfaces
- A clever interface may decide to merge surfaces together that are adjacent and share the same composition - so what is reported is not the same as the user defined.
- Interfaces may decide to subdivide surfaces or components so that the model includes entities that the user did not ask to be created.
- The user may define a compact description which the tool expands into hundreds of lines of description and provides only limited facilities for exploring the expanded entity list.

Some software (including ESP-r) provides both information on the screen and model contents reports. This allows those with graphic interpretation skills and those with report interpretation skills to work together.

A good model contents report would not only be human readable but reflect *any change* to *any entity* within the model. Simulation tools are imperfect and thus some entities may not be reported or may not include a sufficient level of detail.

Ideally, tools should allow the user to define the level of detail for various reported entities as well as which topics to include. Similarly, interfaces should allow the user to expand or collapse interfaces so as to focus on specific issues.

If vendors mount training courses which pay little attention to topics related to model quality and your team considers this important then talk to your vendor and look for alternative training providers. Better yet check before attending the course or buying the product.

Another shortcoming that is attributable to the design of the simulation tool is the ease by which independent actions by multiple users cause clashes which are difficult to resolve. If a model is working and some action by another person causes a fault, confidence can be badly affected. This constrains the deployment of simulation in distributed work groups.

Of course poor working practices can reduce the efficacy of tool facilities. Failure to check for site orientation prior to undertaking shading analysis can waste valuable time just as failure to notice that fire doors may be installed with fail-safe closing mechanisms (that allow mixing between zones) can alter assumptions about ventilation in buildings.

A tool might be just clever enough to name the 27th surface in the zone which is vertical *Wall-27*. The author of the model knows the surface is the *partition_to_corridor* and probably knows several other attributes of the surface. The interface presents the initial guess for editing, five seconds of typing would make this clear. Accepting the default forces everyone else has to mentally reconstruct an understanding from the shape and location of the surface and what it is composed of each time it is selected from a list or

included in a report. The *Cookbook* quote of the day is:

> Names are the first step to understanding and essential to owning ideas.

Of course tools may be imperfect in their implementation of this concept:

- Some interfaces (like ESP-r) constrain the length of entity names. Terse names are a frustration and occasional source of error.
- Some interfaces assign names automatically and do not allow them to be changed - unique but arbitrary names can be opaque to the user.
- Some interfaces do not allow users to name entities within their models - this is unforgivable in terms of the resources required for model checking.

Knowing that simulation tools limit our ability to create self-documenting models it is for the community of users to evolve working practices which compensate for such limitations.

15.2 Responsibilities within simulation teams

Decisions made by team members as they plan and build models can influence the resources needed by others to understand both the intent and the composition of models. Some models *tell a good story* and thus we can quickly use our newly acquired understanding. Other decisions result in models with impose a considerable burden on the design team. Making clients work hard is NOT a profitable business strategy.

When the client arrives and the model is opened up on the computer screen is there recognition, perhaps after a quick tour, and then a move to substantive issues. Or are there puzzled expressions and the meeting gets hijacked by explanations of how the image on the screen represents their design.

This is not an argument for a literal translation of CAD data into a thermal model. Much that is included in CAD drawings is simply noise in the thermal domain. A client who saw a simulation model with fifty thousand surfaces would be justified in asking for the methodology behind this approach to abstraction just as if they were told a box represented the Guggenheim in Bilbao.

Consider what simulation looks like from the clients perspective and choose, if possible, design the model to limit their confusion. It may save time.

Second *Cookbook* quote of the day:

> Self-administered QA brings great joy to someone else's lawyer.

Simulation teams of one are viable in the long term only if they outsource model checking. In a team it should be standard practice to outsource the task to another team member after initial checks are made by simulation staff.

Changes in staffing can require that models be passed to others in the team for completion. Models risk become orphans if the author of the model works in isolation or uses a different style. A model that is opaque to someone in the team is likely to be even worse for a client.

The task of taking another persons model and understanding it well enough to make modifications to it is a classic test of working procedures. Projects go quiet for weeks at a time. If a substantial resource is given over to getting back to speed on a dormant project this could lessen the resources available for other tasks. It does not help that the design of simulation software rarely takes into account that many teams are working on several projects simultaneously.

Model quality issues are different for each participant in the simulation team.

Team manager

The team manager has in interest in ensuring that models are fit-for-purpose and that staff are working within their limits (and the limits of the tool). A model which tells a good story is one which the manager can more easily browse. And managers who regularly review models can anticipate possibilities as well as notice deadlines slipping.

Quality manager

Just as the author of a book needs an editor to help complete the story, neither the team manager nor simulation staff are in a good position to carry out this task.

The quality manager also has an interest in regular reviews of the work in progress as well as ensuring that the model continues to be fit for purpose. Quality managers quickly recognise models which tell a good story from those that do not. Good working practices ensure that simulation staff get this feedback regularly.

Even with good working practices errors become embedded within models. Some typographic errors that get past tool checks will evidence themselves in the predictions of performance *if we are paying attention*. A 10KW casual gain in a room which is supposed to have 2KW may not show up as a temperature difference if the environmental control has an oversized capacity. The review would have to notice the reaction of the environmental control for this zone differed from other similar spaces. If the only report generated was a total for the building then this change might not be noticed.

Some errors can be subtle - selecting the wrong type of glazing for one window out of a dozen windows might alter performance of the room only slightly. The logic of a control which fails for an infrequent combination of sensed conditions may be difficult to spot.

The simulation team should consider the frequency and types of error can exist without altering the patterns of performance to the point where a different design decision is made. Checks by simulation staff are likely to spot some types of errors but others only become apparent by their impact on predictions.

There are benefits in the quality manager being pro-active in the project and using simulation tool facilities to review models, generate model contents reports and review performance predictions. Such investigations need not be a burden in time or computing resources. The skills necessary for

using the tool to carry out these tasks can be acquired in a few days.

Another pro-active task is to work closely with others in the team to ensure that model quality is a continuous part of the model planning and creation process. The quality manager might also devise task sequences to confirm the skills level, strategies and *inventive resources* of simulation staff.

The definition of the quality process used in the *Cookbook* places emphasis on identifying opportunities as the work progresses and here a pro-active quality manager can take a primary role. The quality manager can also be a champion for reserving project resources for the exploration of value added issues.

Simulation staff

The traditional deployment of staff involves junior staff *primarily working* to create models, run assessments and extract performance data. The critical issue for model quality is self restraint. Accepting default names for entities saves a few seconds but requires that others expend effort each time they browse through the model or look at performance reports.

Simulation staff are continuously making decisions and assumptions. For example - a quick decision to go with the standard concrete floor thickness in the database rather than confirming the actual thickness in the model might be valid approach at the time. It becomes problematic if it persists and the decision is forgotten.

Unconstrained keyboard skills of novices can wreck havoc. Novices need active support from others who have more evolved opinions about the thermophysical nature of buildings and systems. If those with opinions take the time to pass on ideas then novices will be better placed co-operate when they start their work.

Topics that experienced users may forget that novices do not know are:

- Information required from the client at different phases of the work
- Benefits and drawbacks of different tools, time and computational resources for different tools.
- Ideas about how design issues might be represented
- Ideas about zoning strategies and level of detail
- Ideas about past models to review

Simulation staff should spend 2-3 hours a week exploring new working practices or generating scripts to automate processes. This investment may result in better procedures and better models.

Domain experts and mentors

The domain expert will likely have strong opinions about they see in performance prediction graphs or reports. Subtle patterns may be easy for them to recognise.

The trick is to get the domain expert to also look at the model and the performance predictions with a view of further tweaks that could be made to improve performance or to confirm that the prediction is in line with expectations.

Mentors can play a pivotal role in enabling staff productivity and fine tuning working practices that ensure clarity in models. Prior experience with similar projects may allow the mentor to be among the first to identify where models are providing unexpected results.

Mentors engaged to explore new topics or explore an alternative way of working will likely be part of the team updating procedures and helping to adapt simulation tool facilities.

Musical chairs

In another section of the *Cookbook* it was suggested that occasional rotation of tasks within a simulation team can be useful. Rotation of quality assurance tasks has several benefits:

- A *fresh pair of eyes* can notice in seconds what has been evading others for hours.
- Conversations needed to confirm *what is this* are instructive for both participants.
- An appreciation how different designs of models impacts model checking tasks can result in better models.

15.3 Model planning

There are many steps between a simulation model that looks like a CAD model and a simulation model that is simplified beyond recognition. The thermophysical nature of a space which is *reasonably represented* by one hundred surfaces is not ten times better if a thousand surfaces are used. And because increasing complexity requires more than a linear increase in resources much thought is required to arrive at an appropriate model resolution.

This said, a small increment in model complexity may provide sufficient visual clues so as to reduce the effort needed to understand a model. Models that clients recognise help them to buy into the process and it often simplifies reporting requirements. This might be as simple as including visual place holders for columns and desks or including a crude block representation of adjacent buildings.

Other aspects of model design are covered in sections 1.4, 2.3, 3.3 and 4.1. In each of these sections the emphasis is on the planning stage and working out ideas in sketches rather than on the keyboard.

Entity names

Names like hmeintflrt_1 might be derived from *horizontal mass element intermediate floor type instance one* but almost no one else will appreciate this baggage. But if a client seems at ease when talking about room 1.12b then that might be the best name for that zone in the model.

Names are especially important where there are lists to select from. For example, if the interface presents a list of 100 surfaces and it takes ten seconds to locate the floor surfaces because of an obscure naming convention and two seconds with a clear naming regime then the pay back is immediate. Selecting the wrong surface can result in costly errors. The *Cookbook* recommends that the first attribution to alter from the default is the entity name and

that names follow consistent patterns within a model. Surface names only need to be unique within a zone but nodes and components within a network need to be unique.

Recording assumptions

As we create and evolve simulation models we make dozens of seemingly trivial decisions and assumptions which soon pass from our memory. If these are not recorded there can be adverse impacts. One of the driving forces for software development is to ensure that the internal data model of the simulation tool has space available to record decisions and assumptions.

The existence of a dialogue for explaining the intent of an occupancy profile does not ensure that it is used. The quality manager has an interest in self-documenting models and working practices should set standards for how decisions and assumptions are recorded.

Placeholders

Entities in simulation models require lots of attribution and the information required may not be available when required. The use of place holders for data not yet confirmed (e.g. creation of an approximate construction in the database) is a valid way of continuing to get work done. Working procedures are needed to ensure that such pragmatic actions are followed up and the model is updated as better information becomes available.

Who does such updates, the frequency of reviews, and decisions about who needs to know that changes have been made are all items to include in a well-ordered working procedure.

15.4 Complexity

The evolution of software now allows simulation teams to create models which are closer approximations to the built environment and these models often involve a level of complexity which would not have been contemplated a few years ago. Our attempts to create better models is tempered by our ability to manage such models.

One of the unexpected problems with software which is designed to be both user friendly and over-functional is the level of self-restraint needed to create models which match the needs of the project. Facilities intended as productivity aids can easily seduce the user into an overly complex models.

Those who are unprepared for complexity multiply their workload as well as the risk that errors and omissions will go undetected. Preparation includes the evolution of working practices in projects of increasing complexity. Another investment is to allow staff to gain confidence in projects of increasing complexity. Clarity in the design of models reduces the attention required for many tasks and is essential within complex models.

Ensuring that the model is correct is, for many practitioners, the critical limit on the complexity of their models. Reviewing a literal representation of a hospital complex with hundreds of rooms to certify the correctness of thousands of entities is, at the least, an iterative task. Each iteration focuses on a different aspect of the model and/or

its performance.

Spot checks are a *useful step* in ensuring the quality of models. Quality managers will develop techniques for scanning model contents reports as well as techniques for reviewing the model form and composition with the tool interface.

A classic point of failure is to allocate time for reviewing model contents but not for commissioning assessments designed to identify semantic errors. Techniques that hi-light semantic errors tend to focus on short period assessments where patterns in the operational regime and boundary conditions should result in expected patterns of response.

Finding expected patterns is usually cause for celebration. The risk is in cutting short our multi-criteria assessments because we are in a good mood rather than concluding the assessments when we have reached a holistic understanding of the design.

In addition to evolving skills and working practices to cope with complexity, successful simulation teams also are able to recognise when complexity can be avoided. During model planning consider whether a design must be represented as one simulation model or whether it can be sub-divided. Subdivision can take two forms - multiple models which combine to the whole of the building and models which contain a selected portions of the building.

There are cases where a fully explicit representation is required - for example in a naturally ventilated building where air flow patterns are widely distributed. In many buildings there is considerable repetition and little to be gained by describing all rooms. The technique requires that the constrained model embody both the typical and exceptional elements of the design. Selecting what can be omitted from a model is a critical step in the planning process just as determining scaling factors is in the calibration phase. And the benefits can be considerable and many simulation groups use such scaling techniques in their projects to conserve the resources needed to create, run and extract data from their models.

Another technique to re-allocate resources for semantic checking is to employ scaling techniques to the assessments carried out. Limited duration assessments (e.g. typical fortnights in each season along with extreme weeks) which are carefully scaled can result in predictions that are very close to predictions of brute-force annual simulations. This technique is especially important for simulation tools such as ESP-r which are disk intensive during the simulation and data extraction phases.

15.5 Multi-criteria assessments

As mentioned elsewhere in the *Cookbook*, time gained by good working practices can provide a reserve of time to explore model performance. The more complex the model the more multi-criteria assessments are essential for discovering unintended consequences of design decisions as well as errors and omissions within the model.

So where might one begin the process of gaining confidence in a model? Well founded opinions as to what should be happening within the building (or

access to such opinions) is a first step. This may take the form of information from similar projects, tabular data in handbooks, access to a mentor or expert or to measurements in this or similar buildings.

The next step is familiarity with tools and techniques for extracting performance data in forms which clarify what is happening within the virtual physics. In workshops for ESP-r, the time allocated for exploring model performance tends to be at least as long as for the model creation tasks. Both interactive investigative skills and the automation of data extraction are part of the process. The following list includes some useful indicators:

- the range of dry bulb temperatures in each zone and the time of the maximum and minimum AND frequency bins to confirm the number of extreme occurrences
- the difference between dry bulb and mean radiant temperature and if extreme do further checks for surface temperatures
- the range of heating and cooling demands and time of peak occurrence AND a frequency distribution of demand
- the number of hours heating and cooling is required AND the number of hours when zones are floating in the dead-band
- casual gains in each zone either as statistics and as a plot to confirm if lighting is switching as expected
- graphs of zone temperatures and zone environmental controls to see if an optimal start or stop might be useful, if there is heat stored in the building overnight or if a modular system might be appropriate
- statistics and graphs of solar entering the zones to confirm that windows that look like they are facing north actually do face north
- when a zone catches your attention check its energy balance for the type of heat transfer associated with big gains and losses

The time of occurrence is mentioned above because a peak temperature during office hours may be very different than a peak temperature when no one is in the building. Frequency bins are mentioned because a peak demands for a dozen hours in a season may not be a good indicator of system capacity and extreme temperatures that are rare may be amenable to demand-side management.

Experts will also re-run transition season assessments without environmental controls or with reduced capacity for environmental controls. Why? Because buildings can often be comfortable with little or no mechanical intervention. The traditional focus on system capacity tends to ignore performance during the hundreds of hours of mild conditions that happen in most regions. A few moments to create a model variant to confirm this can result in significant value to the client.

Another trick of the experts is to gather statistics for the occupied period as well as at all hours. Why? Because attention to after-hours performance provides clues for improving the design of the building and its operating regime.

Some simulation tools such as ESP-r can be driven by scripts to automate the recovery of the data mentioned above. There are two common approaches:

- recording the keystrokes used during an interactive session into a scripts to automate subsequent checks
- defining an Integrated Performance View (IPV) for the model and using this to invoke specific assessments and recover the multi-criteria data.

The use of scripts is covered in an Appendix of the *Cookbook* as well as in the *Automation* section in a subsequent page of this chapter. Setting up an IPV is a topic not yet included in the *Cookbook*.

Working at the edge

Most experts plan their work and their models to avoid the computational and complexity limits of their tools. A failure to anticipate the likely course of the project is a classic way to run short on options.

ESP-r is compiled with specific limits of model complexity and it is worth checking what the current limits are during the planning stages. It may be necessary to re-compile ESP-r if you want different limits (there are alternative header files available for different model resolutions that have been tested). The *Install Appendix* provides information about this. Some simulation tools are written to allow models to grow in complexity without having to re-compile, and there are still limits that knowledgeable users avoid.

Models which have been extended towards the limits of the simulation tool reach a point where productivity can suffer and where dependencies within a working model can be broken and errors introduced. There are many possible vectors for this, some involve actions by the user and some can be gaps in the logic of the software.

Caution and paranoia are useful attitudes. Backup the working model as a first step. Generate a full model contents report and extract a range of performance data for use in comparisons with the revised model. Talk to others about their experiences of managing complex models and for advise on where simplifications can be made *before altering* the model. Plan the sequence of tasks, make frequent backups, carry out calibration runs on the revised model and check these against the initial performance predictions.

In addition to the overhead of working with large models there is a cost in computing time, generation of large performance data sets as well as the time needed to recover performance data. Each simulation tool and data extraction technique tends to have a point where the burden becomes noticeable. Users dependant on Excel for data presentations typically test the limits of column and rows in tables. XML documents tend to become slow to parse and graphs can become unreadable.

The ESP-r is disk intensive. Data extraction gets progressively slower as result files approach one Gigabyte and can be unstable after that point. A large project with a dozen design variants and five minute time steps can take a morning if not a weekend to process.

Thus, the design of the model and the design of the assessments and data recovery tasks are topics to be considered in the planning stages of the project.

Some projects may require a compute server or several workstations to provide timely information to the design team. The option of larger disks and more memory can help reduce the time for data recovery tasks. To see which approach is most appropriate carry out tests on an existing model of similar complexity.

Entropy is a word that begins to describe parametric studies. So much is invested in setting up a base case model that can be adjusted to reflect the various design variants. If after testing and running the scripts the whole edifice can be destroyed by demands to alter the base case and re-run the assessments.

ESP-r has the flexibility to be used for parametric work but the distributed file structure can make it difficult to make global changes to dozens of models and the potentially hundreds of files that support such models.

Automation

Using a simulation tool interface to apply multiple changes to a model can be frustrating. Some users *hack* their model files and some use scripts to perform parametric changes to models. While such actions can save time they are also a potential source of subtle errors if dependencies are not resolved.

Some practitioners make fewer errors than others when creating model variants. A critical review of the order of the tasks and the checks that were done to limit errors resulted in a *create model variant* facility within ESP-r.

As seen in Figure X, the user who wants to create a model variant selects topics from a list and the code determines the related dependencies and manages the model files so that subsequent changes do not corrupt the initial model.

ESP-r includes functions to search and replace instances of a specific construction within the model or to alter attributes of surfaces associated with a named list (also called an anchor list). There are also functions to rotate or transform the model which requires that a number of dependencies be managed. Other global changes to models may require editing of model files by a sequence of manual tasks or the use of scripts.

Experts tend to test their scripts on a duplicate of the model or a portion of the model prior to use. In all cases the working procedure should insure that the altered model is scanned by the tool to see if errors are detected. It is also a good idea to generate a new model contents report and compare that with the initial model.

Some parametric changes require more than a substitution of key words and names in model files. For example, removing a surface requires that many relationships within the model be re-established. It also results in calculations for shading patterns and view factors becoming obsolete.

Simulation tools are designed track these dependencies and attempt to resolve them. A user generated script

might have difficulty duplicating all of the tasks. Consider whether the script might be re-written to invoke the tool interface to carry out these tasks.

In many cases the logic within the tool can be generalised so that the same action that is carried out interactively can be driven from an alternative interaction or command line directive to the tool. Because scripts can be fragile and difficult to maintain it is worth checking with the software vendor to see if it is possible to revise the tool to support such modifications. For open source tools such as ESP-r there are many options for evolving the code to accomplish standard tasks.

15.6 Semantic checks

Ideally we design models to be no more and no less complex that is required to answer the current design question. If we are clever our model will also be designed to answer the next question the client will ask. A model which is overly simple may fail to represent the thermophysical nature of the design. A model which is overly complex absorbs resources which may not be sustainable.

Models which have evolved many times in response to changes in design focus may include details which are no longer relevant or at an in appropriate level of detail. A critical review may be in order to determine if a fresh start will be more productive than further revisions.

Some semantic checks can be carried out by reviewing the client's questions and considering the underlying physics involved. If, for example, the design is sensitive to the distribution of solar radiation within rooms then a review would usefully look at the pattern of surface temperatures during the day vis-a-vis the distribution of radiation.

A design goal to make best use of solar energy for heating while controlling summer overheating will likely require careful attention to geometric resolution and the constructions which are used. To find out if additional geometric resolution (e.g. subdividing walls and floors) yields better predictions then the same room can be represented by three zones at different resolutions. What is learned from this exercise can be applied to the full scale model.

If control of blinds was also included then switching patterns would also be checked. But in this case it may be necessary to compare against the same room without the blind control. Indeed, to clarify the impact of ideal zone controls or system component control a variant model without the control or with a simple control is often the most efficient approach and working procedures should ensure that such model variants can be quickly created or are maintained as the work progresses.

With a general tool such as ESP-r there usually several possible designs of the model. In the planning stage a series of constrained models of different approaches can be created to identify the most promising approach. Some simulation teams will have a range of test models available or will have procedures for setting up test models. The *working practices* chapter recommends that simulation staff have time allocated to exploring alternative

approaches and creating models for testing likely issues that will arise in the design process.

An example enabling a suite of models to test sensitivity is the sequence of models in the *technical features* group of exemplar models distributed with ESP-r. These represent the same pair of cellular offices and a corridor at different levels of resolution and with different simulation facilities enabled.

A semantic check may also be triggered by an unexpected performance prediction.

- Why is this room several degrees warmer than expected?
- What is the largest heat gain to the space? When is it happening?
- What else is happening at the same time or just before this?

Professor Joe Clarke introduced the idea of *causal chaining of energy balances* in the early 1990's. It involves identifying the dependencies implied by a thermal event and working back down the dependency tree to isolate the form of energy transfer, which if corrected, will alter the initially noticed thermal event.

If the initial model was well designed there is a tendency, when the client poses a new what-if question, to assume that the model can be adapted. Sometimes this is the case. And sometimes the new question implies a different domain of assessment or a different level of resolution.

This is a critical semantic check. The resources required for a not-quite-fit-for-purpose model can be similar or greater than the cost of designing and implementing2 a model for the new question. And this is also a semantic check that should be undertaken for broad parametric studies. A critical review may suggest that a higher resolution is required for some portions of the design matrix.

An example is the quality of light in rooms. Generating daylight factors to roughly assess whether the back of a room will be perceived as dark is a very different question than *is there glare at the head of the conference table*. Daylight factors are much less sensitive to the geometry of the facade than the requirements of a glare assessment. One is essentially independent of time and the other is both position and time dependant.

Hardware and human gremlins

> Think of simulation models as prisoners of war which have a duty to escape (or at least ruin your weekend).

There are countless ways for models to be rendered unusable because of corruption, inconsistencies or lost files. Phrases like - *I was going to backup this evening* compete with *I was only trying to make the model work better* and *there was a power spike* become part of the folk history of every simulation group. The intensity and duration of the disruption which follows is determined by the robustness of our working practices.

Making backups is not rocket science. It is amazing how a delayed holiday brought on by a disk crash can change work habits to the point where the loss of a half hours work is considered a

failure within the group.

For every type of computer and operating system that ESP-r runs on there are utility applications available to create archives of model folders. In Unix/Linux/OS X systems commands like `tar cf broxburgh_18_apr_contol_a.tar broxburgh` will put all the files and sub-folders of broxburgh into an archive file. On Windows it is usually a matter of a right click on the model folder to create a zip file. Each simulation group will have its own naming scheme for such files as well as rules as to where these files are stored.

The frequency of backup depends partly on the imagined hassle of repeating a particular set of actions as well as the state of the model. The following are classic trigger points for making backups:

- when a zone becomes fully attributed
- prior to making a model variant
- prior to a search and replace operation
- prior to a transform or rotation
- prior to a change in control strategy
- to record the current state of the model prior to a review
- when someone with electrical test gear is seen in the building

And backups can be focused on one or two files which are related to an issue being tested. A quick test of reducing the cooling capacity in one zone of the building would start with making a backup of the current control, altering a few values in the control definition, running a test and then recovering the initial state of the control. Different groups adopt different styles of file naming conventions and the logs of the actions taken and reverted.

One issue which is much debated is whether model archives should include results files. Some groups include these, some compress such files prior to archiving and some groups include in the archive scripts and directives needed to re-generate results files. ESP-r models can include pre-defined *simulation parameter sets* and many groups build their working procedures around such facilities.

15.7 Team Checklists

The following table includes some of the issues to notice when working with simulation models. Roughly, the order begins at the start of the work. Clearly there are scores of topics which could be added to this table.

The form of the table is to pose a general issue in the left column and provide a list of associated topics or questions in the middle column and related actions in the right column. The *Cookbook* uses terse phrases and only includes a sub-set of related actions within these tables. Consider this as a starting point and create verbose versions of the tables in this chapter and extend the topics as new situations arise and as working procedures for speculative future work is contemplated.

Question	Related issues	Actions
What is interesting about the site?	What do we know about the site? Does the client realise the importance of the site? Is the north arrow on the site plan? How might site issues constrain the design? How might site layout be used to improve the design?	Check the site plan and maps of the area. Find out which way is north. Review site photographs or visit the site. Check site solar and wind obstructions. Check principal wind direction and whether vernacular architecture takes this into account. Collect local opinions of the site and ensure that these are taken into account in model planning. Discuss with design team options for adjusting the building on the site. If orientations may vary ensure naming regime avoids confusion (e.g. avoid north_entrance).
What weather data is appropriate for the site?	What climate data is available near the site? What patterns of weather will impact the building design? How have other buildings adapted to the local weather? What are the design teams ideas about the influence of weather?	Review climate data to establish seasons, typical periods and extreme periods. Review climate data for useful building failure test sequences. Review climate data for duration of extremes as well as the frequency of moderate weather patterns. Discuss with design team the results from initial review of weather.

Question	Related issues	Actions
Is the design evolving or static?	What drawings or sketches can we access? Are there plans, sections and elevations available to review? Are the drawings likely to change? What is the design team debating now and what are likely future topics?	Review drawings or CAD files and gather comments from simulation staff and quality manager. Check that the drawing scale is reasonable and if details might influence the design of the model. Begin planning tasks and discuss the level of detail required within the team. Arrange a meeting with the design team and present sketches of planned model and likely approaches.
What is the composition of the building? Are the goals of the project consistent with these constructions?	Are construction details known? What what-if questions relate to constructions? Are these typical for this building type? Do we have data or sources for missing data? What criteria for selection or rejection of constructions?	Review standard databases for matching constructions and/or SIMILAR constructions. Create place holder constructions and document. Review sections and discuss with design team. Discuss support for selecting appropriate constructions.

Question	Related issues	Actions
How is the building used? Is the building for a known tenant or is it speculative? How are rooms likely to be occupied at different times and seasons?	Does the client have an well-formed opinion about the building use? Is building occupancy an issue for future-proofing? Is there seasonal diversity as well as daily diversity?	Check the assumptions made by the client. Quantify likely scenarios. Consider diversity, holidays and peak use occurrences. Check past models for similar patterns.
What is the clients big idea? What kind of model would confirm this idea? What are others in the design team interested in?	What performance issues are related to the big idea(s)? What analysis domains and metrics of performance will confirm this? Can beliefs held by the design team be confirmed?	Listen to the language used and review sketches to identify beliefs behind the big idea(s). Identify issues and determine if they can be assessed. Review available tools to check for matches in numerical capabilities and reporting facilities. Brief the design team on the approach to be taken and the likely information to be derived from the assessments.

Tool check checklists

Tool checking will typically include issues such as *is this polygon flat* and *does the construction attribute point to a correct entity in the database* and *does the boiler have a non-negative capacity*.

Tool checks are not likely to notice is a surface named *door* which is composed of *150mm of concrete* or which is *horizontal*. It is unlikely that this combination is true but it can happen and users are in the best position to notice this. *Fire-access-door* would be a better name in this situation.

Scanning of the model done by the simulation engine can include topics that are not checked as the model is created or standard reports are generated. With most simulation tools is it possible to pass the model to the simulation engine without actually running a full simulation.

If that scan passes that then running the standard calibration assessments usually only takes a few moments. If there is a standard script for extracting predictions covering several performance criteria then that step also takes little time.

Indeed it is often a trivial task to automate many syntax and semantic checks freeing up time for review of the output. The generation of automation scripts is a valuable skill within in a team and a consideration for selecting simulation tools.

The criteria for determining the frequency of checks should be included in the working procedures. Typically these would happen at several points during the evolution of the model as well as after simulations have been run. Usually it takes only a few seconds to commission a check but much longer to review the reports and predictions so a balance must be found.

There are techniques for reducing the time taken. Incremental checks are well supported by looking at the differences against an earlier report and software for hi-lighting the differences between two files or between two folders is readily available. The use of pattern matching tools can identify key words in model contents reports.

Ideally the assessment would be designed to test a number of attributes of the design that signal whether its performance is as expected or exceptional.

The table that follows illustrate some of the issues during the use of tool checking facilities. Use this as a starting point for generating your own responses to issues as they arise and for planning future tasks.

Core question	Related issue	Actions
What changed since version 1.3?	Who is working on the model? What tasks are they involved in? Does this match the work plan?	Consult the work log or the log of actions generated by the tool. Generate a model contents report and see how this differs from the previous report. Check with the quality manager and simulation staff about current status.
The client wants to test a different roof type	Which parts of the model are affected? Is information available on the new roof? What performance issues might change? What tasks are required and which staff should be involved? Is this a model variant or an edit of the existing model?	Update the constructions database if required. Generate a model contents report, archive the current model and performance predictions. Create model variant if required. Apply changes, generate model contents report, check against previous report, re-run calibration assessments and review. Re-run standard assessments review. Archive the revised model and brief the client.
The interface says 'problem edges' in conference zone.	Is there a missing surface? Are there edges which do not follow the rules used in the tool? Is there one or more reversed surface?	Look at the wire-frame for missing surface labels. Use the check vertex topology option for a report. Turn on surface normal arrows to review orientation of surfaces. Check for vertices linked to only one surface.

Core question	Related issue	Actions
The ceiling void is warm and retaining sufficient heat to cause a morning cooling issue.	Under what operating regimes and weather conditions is this happening? What are the primary gains within the ceiling void? Is it possible to remove heat from the ceiling void?	Carry out checks under different weather patterns. Review the energy balance in the ceiling void. Review how boundary conditions are represented and the state of adjacent zones. Test different operating regimes to determine sensitivity. Force a brief purge of heat from the ceiling void and check the whether (and how long) it takes the condition to re-establish.

15.8 Simulation outputs

In simulation the time when the *fat lady sings* is most often after the simulation has been run and the team is exploring the performance predictions looking for *the story* to tell the rest of the design team.

Some simulation groups generate the same report irregardless of the project. Such poor value for money is largely the fault of the client for not making their needs clear or not having sufficient experience to know the variety of deliverables that a simulation team can provide.

It is only natural that simulation staff have their preferences for the graphs and tables that are included in reports. However, the goals of many projects are multi-criteria and working patterns need to be established to ensure that a range of performance issues are tested and taken into account in reports to the client.

Gremlins enjoy watching us find patterns we expect, declaring success and then having someone else discover the chaos. The risk of unintended consequences can be reduced by multi-criteria assessments and by different people with different agendas attempting to understand the predictions.

This is not a new issue. ESP-r has long included the concept of an Integrated Performance View (IPV) where a number of performance issues are identified at the planning stage and recorded in the model so that recovery of multi-criteria assessments can be automated. A typical range of metrics would be comfort, system capacity, energy use over time, emissions of carbon dioxide, distribution of light in rooms and the number of hours of system use. As implemented, the IPV is imperfect and further work is required to allow risk to the better managed and opportunities identified.

In other simulation tools this could be implemented by the inclusion of 'performance meters' for a range of issues. The critical step is taking the time early in the design process to identify issues which would confirm issues which may arise as design options are tested.

The table that follows is a sample of issues and questions and actions for staff attempting to understand performance predictions. Use this as a starting point for your own working practices.

Core issue	Related issue	Actions
Is the model still calibrated?	Have calibration assessments been run? Were there enough measurements to understand performance?	Review the criteria for calibration assessments. Identify measurements which match expectations. Identify unexpected measurements and investigate further.
Is performance tracking planning stage expectations? What issues were mentioned in recent meetings? What value added issues are under consideration?	Does the data recovery script include these issues? Do we have trigger values for failure? What other data views will complement standard reports and graphs? Who can confirm the additional reports?	Explore performance interactively, including a few tangent issues. Get a second interactive review. Follow up issues by looking at dependant or related issues. Run test script to confirm data recovery logic and reporting format is ok. Do spot checks. Run the full script. Do spot checks and get someone else to confirm.
What if we altered the vision glass?	What products would be likely candidates? Do we have thermophysical and optical data? What criteria would signal better performance? Is performance sensitive to the facade orientation?	Establish what is available and the claimed benefits. Review data sources or generate optical data Identify glazing-related improvements Plan what to measure and where alternative glazing could be applied. Implement design variants. Archive model, establish base case performance, test substitution and data extraction procedures. Implement one change at a time and compare against the base case. Rank the design variants and identify performance changes. Get a second opinion.

15.9 The model contents report

This section is a review of the ESP-r model contents report. What should you be looking for? What are key words to scan for? How much detail is available? Where is is information held? The reports generated by other tools will differ in detail but they will cover many of the same issues. Those of you who are constrained to screen captures of the tool interface can also create rules for interpreting entities shown in the interface.

The following listings are portions of the model contents report for the doctors office used in the initial chapters of the *Cookbook*. Prior to each section is a discussion of the contents of the report. After the report fragment is a discussion of key words and phrases to look for.

The user has the option to selected which topics are included in the report as well as the level of detail included for each topic. The example below used the verbose level for most topics.

As mentioned elsewhere, many quality managers will create a hard-copy of such reports and use them in conjunction with the interface when reviewing models. One common pattern is for simulation staff to generate the report and then insert additional notes and assumptions or highlight blocks of text that require further discussion and pass this to the quality manager along with the location of the model.

The header

The header of the model report focuses on site and high level project related information. Some of the text is based on user supplied phrases and other parts are generated from model data and file names.

The key words for this section are the date that the report was printed and the name of the model log file (which may contain useful additions by the user).

The summary of the site location and the climate should be checked to see if they match the project. Finding default values may be indicative of a lack of attention (quality managers will notice this).

Lines identifying the building and the building owner and the simulation team have not yet been filled out.

The first two lines of the report also are the place to identify what is different about this model. This can be important if there are several model variants - users in a hurry can easily select the wrong model for a task!

The year mentioned in the summary will have been initially set to match that of the climate file. One reason to alter the year is to shift the day of the week - for example, in 2001 the first of January is a Monday.

```
This is a synopsis of the model Case-study of the ESP-r cookbook: 2 zones;
reception + examination defined in doctor_office.cfg generated on Fri Aug  1
14:21:25 2008. Notes associated with the model are in doctor_office.log

The model is located at latitude   52.40 with a longitude difference of  -1.73
from the local time meridian. The year used in simulations is 1995 and
weekends occur on Saturday and Sunday.
The site exposure is typical city centre and the ground reflectance is 0.20.

Project name:  not yet defined
Building address:  not yet defined
Building city:  not yet defined
Building Postcode:  not yet defined

The climate used is:  and is held in:
/usr/esru/esp-r/climate/uk_birmingham and uses hour centred solar data.
```

The databases

The next section of the report focuses on databases associated with the model. Some of these are standard databases (the path (/usr/esru) is the clue) and some are model specific (a path ../dbs).

Databases in the standard esp-r/databases folder are initially supplied with the software. They might have been expanded or details modified for the group. Other databases can be added to this folder. It is up to the group to manage and document the databases. Databases items should be reviewed and the contents *should* be of a know quality and kept up to date by the quality manager.

Databases associated with the model may or may not be derived from the standard database. Typically they will include items specific to the model. Such entities may eventually migrate to a standard database. It is up to the simulation team to supplement the information held in the model data files with additional notes on the new entities.

Each database differs slightly in implementation. Currently the constructions database contents are the core of the report. The constructions make reference to items in the materials database and if the construction is transparent to the associated optical properties. The layout of the report is similar to that used in the interface.

The constructions part of the report is a combination of numbers and labels which lacks clarity for some users. It follows an older style of file with an emphasis on space for data and limited documentation. The name of the construction is restricted to twelve character. This constraint reduces the clarity of the report.

Until the data structure is revised it is up the user to compensate for the limitations in the tool. No doubt other simulation environments have limitations. The use of additional documentation helps but it relies on persistence by the user community. What is required are clear channels of communication between the user community and and software vendors and mechanisms for converting user demands into changes in software.

```
Databases associated with the model:
 pressure distributions : /usr/esru/esp-r/databases/pressc.db1
 materials              : /usr/esru/esp-r/databases/constr.db2
 constructions          : ../dbs/doctor_office.constrdb
 plant components       : /usr/esru/esp-r/databases/plantc.db1
 event profiles         : /usr/esru/esp-r/databases/profiles.db1
 optical properties     : /usr/esru/esp-r/databases/optics.db2
  . . .
  sections skipped
  . . .
Multi-layer constructions used:

 Details of opaque construction: extern_wall

 Layer|Prim|Thick |Conduc-|Density|Specif|IR  |Solr|Diffu| R    |Descr
      |db  |(mm)  |tivity |       |heat  |emis|abs |resis|m^2K/W
 Ext     6  100.0     0.960  2000.   650. 0.90 0.70    25.  0.10 Lt brown brick
    2  211   75.0     0.040   250.   840. 0.90 0.30     4.  1.88 Glasswool
    3    0   50.0     0.000     0.     0. 0.99 0.99     1.  0.17 air  0.17 0.17 0.17
 Int     2  100.0     0.440  1500.   650. 0.90 0.65    15.  0.23 Breeze block
 ISO 6946 U values (horiz/up/down heat flow)=  0.393  0.397  0.387 (partition)  0.379
 Total area of extern_wall is     92.70
 . . .
 Details of transparent construction: dbl_glz      with DCF7671_06nb optics.

 Layer|Prim|Thick |Conduc-|Density|Specif|IR  |Solr|Diffu| R    |Descr
      |db  |(mm)  |tivity |       |heat  |emis|abs |resis|m^2K/W
 Ext   242    6.0     0.760  2710.   837. 0.83 0.05 19200.  0.01 Plate glass
    2    0   12.0     0.000     0.     0. 0.99 0.99     1.  0.17 air  0.17 0.17 0.17
 Int   242    6.0     0.760  2710.   837. 0.83 0.05 19200.  0.01 Plate glass
 ISO 6946 U values (horiz/up/down heat flow)=  2.811  3.069  2.527 (partition)  2.243

 Clear float 76/71,     6mm, no blind: with id of: DCF7671_06nb
 with 3 layers [including air gaps] and visible trn: 0.76
 Direct transmission @ 0, 40, 55, 70, 80 deg
   0.611 0.583 0.534 0.384 0.170
 Layer| absorption @ 0, 40, 55, 70, 80 deg
    1  0.157 0.172 0.185 0.201 0.202
    2  0.001 0.002 0.003 0.004 0.005
    3  0.117 0.124 0.127 0.112 0.077
 Total area of dbl_glz is      9.30
```

When reading the report remember the layers go from 'other-side' to room side. Make a point of comparing the reported U values with other data sources to ensure that the construction is properly defined. In the case of ESP-r, a U value is a derived value rather than an integral part of the calculation process. It is also up to the user to correctly account for the changes in heat transfer between the centre of glazing and sections near the frame. There are several approaches such as using averaged values and using separate surfaces for the centre and edge of glass but little consensus in the community.

Constructions which are transparent have additional entries in the report for optical properties. This part of the report includes jargon and a not particularly human readable format. The data used by ESP-r is somewhat more detailed that that provided by some manufacturers. And it does not include some data generated by optical analysis tools such as Window 5.2 and WIS. ESP-r is not unique in simplifying some aspects of diffusing glass and advanced light re-directing glazing products.

Zone controls

The next section of the model report is extracted from the controls defined for zones and/or flow or system networks. The documentation phrases are provided by the user. Next there is a short synopsis of what is being sensed and what is actuated. This is followed by an automatically generated synopsis for each period in the control schedule.

The control section of the report is a compromise. The space available for translating the parameters of each period's control law is limited. Some controls include more than a dozen parameters and some abbreviations are necessary. Some controls include jargon used by control engineers. There are a few controls which have almost no translation of the control parameters. Some controls which have parameters which include auxiliary sensor definitions can be confusing because the standard report of sensor and actuator locations does not know that it has been superseded.

Most users take some time to get used to how ESP-r represents controls. Further practices is required to learn how to spot errors and omissions in the control report. Often a close inspection is only done if the model is not working as expected.

This model is probably at an early stage. The intent of the controls is not clear. There is no mention of why 2KW capacity was specified for heating and cooling. In many groups this would be a substantial breach of protocol. An alternative approach would be to provide extensive documentation about what was defined in the planning stage and then gradually improve its implementation.

Documentation is important for ideal controls because the sensor actuator control law pattern used by ESP-r can approximate any number of physical devices. It may be that a later version of the model shifts from ideal representations to component based representations of environmental controls and initial statements can assist this transition.

```
The model includes ideal controls as follows:
Control description:
A basic control is used during occupied periods on weekdays with a set-back
in the evenings. Saturday and Sunday is allowed to free-float.

Zones control includes  1 functions.
An idea control with a dead-band of 4 degrees is used.

 The sensor for function  1 senses the temperature of the current zone.
 The actuator for function  1 is air point of the current zone
 The function day types are Weekdays, Saturdays & Sundays
 Weekday control is valid Sun-01-Jan to Sun-31-Dec, 1995 with  3 periods.
 Per|Start|Sensing  |Actuating  | Control law    | Data
   1  0.00 db temp   > flux      basic control    2000.0 0.0 2000.0 0.0 10.0 30.0 0.0
basic control: max heating capacity 2000.0W min heating capacity 0.0W max cooling
capacity 2000.0W min cooling capacity 0.0W. Heating setpoint 10.00C cooling setpoint
30.00C.
   2  7.00 db temp   > flux      basic control    2000.0 0.0 2000.0 0.0 20.0 24.0 0.0
basic control: max heating capacity 2000.0W min heating capacity 0.0W max cooling
capacity 2000.0W min cooling capacity 0.0W. Heating setpoint 20.00C cooling setpoint
24.00C.
   3 19.00 db temp   > flux      basic control    2000.0 0.0 2000.0 0.0 10.0 30.0 0.0
basic control: max heating capacity 2000.0W min heating capacity 0.0W max cooling
capacity 2000.0W min cooling capacity 0.0W. Heating setpoint 10.00C cooling setpoint
30.00C.
 Saturday control is valid Sun-01-Jan to Sun-31-Dec, 1995 with  1 periods.
 Per|Start|Sensing  |Actuating  | Control law       | Data
   1  0.00 db temp   > flux      free floating
 Sunday control is valid Sun-01-Jan to Sun-31-Dec, 1995 with  1 periods.
 Per|Start|Sensing  |Actuating  | Control law       | Data
   1  0.00 db temp   > flux      free floating

 Zone to control loop linkages:
 zone ( 1) reception    << control  1
 zone ( 2) examination  << control  1
```

Zone composition

The next section of the model report provides a a summary and then a detailed view of zone composition. The summary provides a quick overview of the size and complexity of each zone. If the attribution in a zone is incomplete the summary will reflect this.

The report for each zone reflects the verbosity selected before the report was generated. Much of the report is based on information derived from scanning the form and composition of the zone. If the zone is fully attributed then U-values and UA values are reported for a quick reality check.

Note that the derived values may be a bit confusing if the zone shape is complex - for example if portions of the facade are horizontal and facing up then they will be included in the roof area. Transparent surface that are not vertical may be reported as skylights.

The list of surface attributes is probably best viewed in conjunction with a wire-frame image of the zone. This is one of the places where a well designed naming regime will allow inconsistencies to be identified quickly. The example included below is for version one of the geometry file. The version 1.1 includes additional attributes.

```
ID Zone        Volume|          Surface
   Name        m^3   | No. Opaque  Transp  ~Floor
 1 reception    120.0  12   165.5   4.5    40.0  reception L shaped room, 2 windows, 2 inter
 2 examination   60.0  10    88.3   4.8    28.0  examination: square room with a sloop roof,
   all          180.   22   254.    9.     68.

 Zone reception ( 1) is composed of 12 surfaces and 26 vertices.
 It encloses a volume of 120.m^3 of space, with a total surface
 area of 170.m^2 & approx floor area of 40.0m^2
 reception L shaped room, 2 windows, 2 internal walls with examin
 There is 94.000m2 of exposed surface area, 54.000m2 of which is vertical.
 Outside walls are 123.75 % of floor area & avg U of 0.393 & UA of 19.439
 Flat roof is 100.00 % of floor area & avg U of 1.799 & UA of 71.970
 Glazing is 11.250 % of floor & 8.3333 % facade with avg U of 2.811 & UA of 12.648

 A summary of the surfaces in reception( 1) follows:

 Sur| Area   |Azim|Elev| surface     | geometry | construction |environment
    | m^2    |deg |deg | name        |type |loca| name         |other side
  1  12.0    180.   0. r_int.wall_1 OPAQUE VERT partition    ||< r_int.wall_1:examination
  2  9.90    270.   0. r_int.wall_2 OPAQUE VERT partition    ||< r_int.wall_2:examination
  3  9.75    180.   0. r_wall_south OPAQUE VERT extern_wall  ||< external
  4  21.0     90.   0. r_wall_east  OPAQUE VERT extern_wall  ||< external
  5  9.75      0.   0. rWall1-north OPAQUE VERT extern_wall  ||< external
  6  12.0      0.   0. rWall2-north OPAQUE VERT extern_wall  ||< identical environment
  7  9.00    270.   0. r-wall-south OPAQUE VERT extern_wall  ||< external
  8  40.0      0.  90. r-ceiling    OPAQUE CEIL roof_1       ||< external
  9  40.0      0. -90. r-floor      OPAQUE FLOR grnd_floor   ||< ground profile  1
 10  2.25    180.   0. r_window_sou DCF767 VERT dbl_glz      ||< external
 11  2.25      0.   0. r_window_nrt DCF767 VERT dbl_glz      ||< external
 12  2.10    270.   0. indoor_door  OPAQUE VERT int_doors    ||< ex_int_door:examination

 An hourly solar radiation distribution is used for this zone.
 Shading patterns have been calculated for this zone.
```

After the surfaces have been listed there may be additional lines which identify optional attributes of the zone. In the example below is a notice that shading patterns have been pre-calculated for the zone.

What is missing from this report are the details of the polygons which make up the surfaces of the zone. Users interested in this level of detail will either have to look at the interface (where coordinates and edge lists are found) or look at the model descriptive files.

There are additional data fields in comparison with the interface so care and attention to the report can help identify issues in the model.

Quality managers will scan this report for 'UNKNOWN' as well as for names that do not match the entity. For example a wall named south_wall might actually be facing West. Is this because the zone was rotated after the surfaces were named? And there is also the common error that a surface polygon has been inverted and what looks correct may be facing the wrong way.

Zone schedules

The next section of the report focuses on the schedules of air flow and casual gains defined for the zone. First there is the user supplied documentation and then the data for each of the schedule types.

The format of the report is similar to that supplied in the interface and thus it is somewhat terse. In the example

below there is no control imposed on the air flow schedule and so no information on the control logic is included.

The current version of ESP-r supports a minimum period of one hour so there can be up to 24 periods in a day. Periods should be in sequence and include the whole of each day type. The interface will attempt to sort the data provided by the user to fit this rule.

Working procedures that minimise errors typically enforce matching documentation and data. Numbers may be correct without being clear as to what they represent.

There is a brief note about the values for the air flow schedule but there is nothing about the casual gains. There does seem to be some diversity included in the casual gains for occupants. Lights are a fixed wattage as are small power loads. Again this is probably early in the design process. A quality manager would notice that the radiant and convective split is 50 50 for occupants and lights and equipment and ask for clarification.

```
 Air schedule notes:
One half air change for all days and all hours (initial engineering guess).
 Control: no control of air flow

 Number of Weekday Sat Sun air change periods =  1  1  1
     Period   Infiltration   Ventilation     From Source
     id Hours Rate ac/h m3/s Rate ac/h m3/s  Zone Temp.
Wkd  1  0 - 24     0.50  0.0167    0.00  0.0000   0     0.00
Sat  1  0 - 24     0.50  0.0167    0.00  0.0000   0     0.00
Sun  1  0 - 24     0.50  0.0167    0.00  0.0000   0     0.00

Notes:
no operations notes (yet)
 Number of Weekday Sat Sun casual gains= 13  0  0
Day Gain Type     Period Sensible  Latent     Radiant    Convec
    No.  labl     Hours  Magn.(W)  Magn. (W)  Frac       Frac
Wkd  1 OccuptW    0 -  7      0.0       0.0       0.50       0.50
Wkd  2 OccuptW    7 -  8     80.0      40.0       0.50       0.50
Wkd  3 OccuptW    8 -  9    240.0     120.0       0.50       0.50
Wkd  4 OccuptW    9 - 12    400.0     200.0       0.50       0.50
Wkd  5 OccuptW   12 - 14    240.0     120.0       0.50       0.50
Wkd  6 OccuptW   14 - 17    400.0     200.0       0.50       0.50
Wkd  7 OccuptW   17 - 24      0.0       0.0       0.50       0.50
Wkd  8 LightsW    0 -  8      0.0       0.0       0.50       0.50
Wkd  9 LightsW    8 - 19    150.0       0.0       0.50       0.50
Wkd 10 LightsW   19 - 24      0.0       0.0       0.50       0.50
Wkd 11 EquiptW    0 -  8      0.0       0.0       0.50       0.50
Wkd 12 EquiptW    8 - 17    100.0       0.0       0.50       0.50
Wkd 13 EquiptW   17 - 24      0.0       0.0       0.50       0.50
```

Users who may have *hacked* their files may have introduced gaps or overlaps in the periods and this report is one way to identify such occurrences.

15.10 Summary

Well-formed working procedures will ensure that models tell a clear story to the design team and that performance predictions have been reviewed to ensure that predictions are within normal expectations and options for better than expected performance are delivered to the design team.

It is a challenge to ensure resources used for generating and testing models is within the project budget and that sufficient time and attention is available for exploring value-added issues. A pro-active quality manager is needed to champion such issues.

The quality of models is a result of decisions made during the planning stage, the design of the model, actions by members of the simulation team and the facilities included within the simulation tool.

The increasing complexity of models is partly driven by new questions from design teams and by productivity features in simulation tools but is constrained by our ability to confirm that models are both syntactically and semantically correct.

The discussions and tables included in this chapter are intended to be a starting point for simulation teams to generate robust working procedures and provide ideas for skills which may be useful to acquire.

14 Install Appendix

ESP-r is available on a number of computing platforms and this section provides information on how to acquire ESP-r pre-compiled distributions or to checkout one of the current ESP-r distributions from the Subversion repository.

ESP-r was initially a suite of tools running on Sun workstations and then with the advent of Linux running on lower cost personal computers the code was adapted to also run on Linux. There are a few lines of code which required adaptation for Solaris and Linux platforms and there are almost no differences in user interactions and in administrative tasks.

ESP-r implicitly assumes a range of operating system services and protections. For example, that corporate databases and example models are held in folders where normal users can read but not overwrite such files. On other computing platforms such protections are either enforced in a different way or absent.

Note that ESP-r assumes the computer environment is using a USA or UK locale and that real numbers use a period as a decimal point and that a comma, tab or space is a separator between data. There is also a restriction that names of entities use an ASCII character set rather than extended character set. These dependencies are related to the underlying Fortran source code read and write statements. ESP-r has been observed to have problems with some, but not all Asian keyboards and locales.

Solaris supports an F90/C/C++ compiling environment as well as the GNU compiler collection. The former is particularly useful for development work as Sun supports IEEE floating point exceptions and array bounds checking.

Linux supports the GNU compiler collection. Note that recent Linux distributions tend to have version 4.2 of the GNU compilers and ESP-r currently works better with the older 3.4 version of the compilers. There is also a general issue with 64 bit computers - some users have experienced difficulties running ESP-r in graphic mode on such computers and ESP-r is thus more robust in 32 bit computers.

With the advent of OSX, Apple computers offer many of the same compilers and low level operating system services as Linux and so it has been possible to port ESP-r to Apple computers.

There are a few minor differences in operating system services (file name case sensitivity is incomplete and users folders are found in /Users rather than /home.

In terms of use the interface is the same as is offered on Linux. Because it does not follow the full OSX look and feel rules, some users find this confusing.

OSX supports the GNU compiler collection as well as X11 libraries and source code conventions. For development work it is necessary to install the so-called *fink* facilities as well as X11 support. A full list of requirements can be found in the ESRU web pages. The pre-compiled distribution is made on a PPC rather than Intel Apple computer. It has been reported to run on Intel based Apple computers.

Because of the differences in compilers and operating system services it took some time to realise a version of ESP-r that runs natively on Windows computers. The initial approach to ESP-r running on Windows computers was to use an emulation environment called Cygwin. Cygwin provides the compilation environment required by ESP-r as well as translating many operating system requests and providing a similar command line interpreter (shell scripting) as one would find on a Linux machine.

Again there few code differences required for developemnt and use of ESP-r on Cygwin. In terms of user experience, ESP-r thinks it is running on a Linux box and the same user interactions apply.

Cyginw supports the usual GNU compiler collection and X11 graphics libraries and the development tasks are essentially the same as on Linux. File permissions are less strict than Linux and thus care should be exercised to avoid overwriting files that ESP-r assumes have strict permissions.

The native Windows version of ESP-r is an *almost complete* port of the facilities available on other computer platforms. This version works on Windows 2000 and Windows XP computers. There has been little or no testing on Windows Vista or on 64-bit versions of Windows.

The underlying graphic libraries currently restrict some functions (this is work-in-progress). The major differences are found in the facilities provided by the operating system and in the layout and conventions of the file system.

ESP-r currently has a limited ability to cope with spaces in file names and it also has limits on the length of file names. These limit where ESP-r can be installed as well as how deeply nested model folders can be before file names become truncated. For this reason, pre-compiled versions of ESP-r are designed to be in `C:\Esru\esp-r` rather than in `C:\Program Files\Esru.` ESP-r models work better in `C:\Esru\Models` rather than `C:\Documents and`

`Settings\Fred\Current Models\`

Development for Native Windows currently requires the MSYS collection of tools in addition to MinGW, a port of the GNU compiler collection.

Environment variables and files

When ESP-r is initially compiled several types of information are embedded in the executables (e.g. where ESP-r is installed) and other types of information (e.g. where to find example models and what databases to intially load) is scanned in from text files. One of these text files is called `esprc` and the standard version is assumed to be in the installation subfolder `esp-r.` Its contents are listed in Figure X and the meaning of the tokens is presented below.

The file is in tag - data, data format. Typically the first token is a label and the second token is either an executable to be invoked or the name of a file to be used. To alter this initial specification use a text editor and change the relevant token as required. Look in the `preferences` menu of the Project Manager to access the details of this file.

The intially created version of the *esprc* file is held in the ESP-r installation folder. If a user wants a custom version of this file to use they should copy it to their *home* folder with the name *.esprc*.

- *ESPRC - this is the file type tag. It must be the first line
- *gprn - commands associate with capturing a rectangular section of the screen. The 2nd token **import** is the executable which captures a section of the screen. In this case the esprc file was created with a Linux computer and the executable name would be different for a different computer operating system.
- *tprn - commands associated with dumping the current text feedback buffer to file will write to the file identified in the second token.
- *gxwd - a variant of *gprn but which captures the whole screen.
- *cad - instructions for a CAD tool to invoke. The second token is the executable and the third token is a key word describing the type of file it creates.
- *image_display - commands related to the display of model-associated images. The second token is a key word identifying the format of the file and the third token is the name of the executable to invoke to display that type of image. There can be several *image_display lines in the *esprc* file.
- *journal - turns on a time-stamp facility which logs user actions and the key words are ON and OFF.
- *editor - which ASCII text editor to invoke if an external application is required.
- *report_gen - not used
- *exemplars - the name of the file to read which includes a list of models which can be accessed and where they are stored. The initial contents of the *exemplars* file is for use in ESP-r workshops but the contents can be edited to include other models.

- *validation_stds - the name of a file to read with information needed to comission standard tests
- *db_defaults - the name of a *default* file which holds a list of initial databases. If you want to use an alternative list of initial databases edit this file or include a reference to an alternative list of databases.
- *db_climates - the name of a *climatelist* file which holds a list of climate data sets and their location. If you want to use an alternative list edit the file or provide the name of an alternative file.

Default file assumptions

The second file which is commonly scanned when ESP-r modules start is the *default* file. The name of this file is included in the *esprc* file. The file is a tag - data format and is typically found in the installation folder. An example of this file is listed in Figure 10.2. Note that the path */Users/jon/esru_prj_dev* points to an installation made for testing purposes and this path was generated as the test version of ESP-r was compiled based on the directives given at the time.

As with the previous files the name of the file is associated with a specific topic and/or dialogue within the user interface. These dialogues associated with specific types of model files require a default name and the default file names are scanned in via the *default* file rather than being hard-coded into the interface. The name of the file can be altered by editing the file.

- *ESP-r Defaults - this must be the initial line of the file.
- *ipth - this is the path to where ESP-r has been installed based on the specific commands given during the installation process
- *cfg - this is a default file name for a model configuration file (useful for demonstration purposes)
- *ctl - this is a default file name for control loop definitions
- *mfn - this is a default file name for an air flow network
- *dfd - this is a default file name for a CFD domain description
- *res - this is a default file name for a zone predictions (results) file. This file should be created during the install process so that it is easy to demonstrate ESP-r.
- *mfr - this is a default file name for mass flow predictions
- *clm - this is a default file name for climate data. This climate file should be created during the install process.
- *prs *prm *mlc *opt *evn *pdb - these are default file names of databases (in case the user request a default database. Many users will change the name of the database files to suite the needs of their work. This file can be accessed via the `preferences` menu of the Project Manger.

```
*ESPRC
*gprn,rectangular dump,import
*tprn,Text dump,/tmp/tx_dump
*gxwd,screen dump,import -window root
*cad,CAD package,xzip,ZIP
*image_display,TIF,xv
*image_display,XBMP,xv
*image_display,GIF,xv
*image_display,XWD,xv
*journal,OFF
*editor,editor,nedit
*report_gen,Reporting tool,xfs
*exemplars,Exemplars,/Users/jon/esru_prj_dev/esp-r/training/exemplars
*validation_stds,Validation standards,/Users/jon/esru_prj_dev/esp-r/validation/s
tds_list
*db_defaults,Defaults,/Users/jon/esru_prj_dev/esp-r/default
*db_climates,climatelist,/Users/jon/esru_prj_dev/esp-r/climate/climatelist
*end
```

Figure 16.1 A typical esprc file.

```
*ESP-r Defaults
*ipth /Users/jon/esru_prj_dev/esp-r
*cfg /Users/jon/esru_prj_dev/esp-r/training/basic/cfg/bld_basic.cfg
*ctl /Users/jon/esru_prj_dev/esp-r/training/basic/ctl/bld_basic.ctl
*mfn /Users/jon/esru_prj_dev/esp-r/training/basic/networks/bld_basic_af1.afn
*dfd /Users/jon/esru_prj_dev/esp-r/training/cfd/template.dfd
*pnf /Users/jon/esru_prj_dev/esp-r/training/plant/vent_simple/cfg/vent.cfg
*res /Users/jon/esru_prj_dev/esp-r/databases/test.res
*mfr /Users/jon/esru_prj_dev/esp-r/databases/test.mfr
*clm /Users/jon/esru_prj_dev/esp-r/climate/clm67
*prs /Users/jon/esru_prj_dev/esp-r/databases/pressc.db1
*prm /Users/jon/esru_prj_dev/esp-r/databases/material.db3.a
*mlc /Users/jon/esru_prj_dev/esp-r/databases/multicon.db2
*opt /Users/jon/esru_prj_dev/esp-r/databases/optics.db2
*evn /Users/jon/esru_prj_dev/esp-r/databases/profiles.db2
*pdb /Users/jon/esru_prj_dev/esp-r/databases/plantc.db1
*ecdb /Users/jon/esru_prj_dev/esp-r/databases/elcomp.db1
*mcdb /Users/jon/esru_prj_dev/esp-r/databases/mscomp.db1
*icdb /Users/jon/esru_prj_dev/esp-r/databases/icons.db1
*mldb /Users/jon/esru_prj_dev/esp-r/databases/mould.db1
*sbem /Users/jon/esru_prj_dev/esp-r/databases/SBEM.db1
*end
```

Figure 16.2 A typical default file.

```
*CLIMATE_LIST
*group  ESRU standard climates
# WARNING: Keep this file up to date with current directory structure !
*item
*name   Default UK clm Climate
*aide   Climate data as distributed with ESP-r for testing purposes.
*dbfl   /usr/esru/esp-r/climate/clm67
*winter_s 2  1  12  3  30  10  31 12
*spring_s 13 3  14  5   4   9  29 10
*summer_s 15 5   3  9
*winter_t 6  2  12  2  20 11 26 11
*spring_t 17 4  23  4   2 10  8 10
*summer_t 3  7   9  7
*avail  ONLINE
*help_start
Location is 52.0N and 0.0E. The solar radiation is Direct Normal.
 Month       Minimum  Time           Maximum  Time         Mean
 Jan          -6.4 @20h00 Sun  8    12.7 @14h00 Sun 29     3.8
 Feb          -1.9 @ 5h00 Tue 14    12.2 @13h00 Thu  2     5.2
```

```
 Mar          -0.8 @24h00 Fri 31    16.1 @15h00 Tue 21     6.8
 Apr          -1.9 @ 2h00 Sat  1    19.4 @15h00 Mon 17     7.1
 May           0.0 @ 3h00 Wed  3    22.7 @14h00 Thu 11    10.4
 Jun           5.0 @ 2h00 Fri  9    21.1 @15h00 Tue  6    13.6
 Jul           9.4 @ 3h00 Mon  3    27.7 @12h00 Mon 17    18.0
 Aug           7.7 @ 4h00 Sat  5    24.4 @12h00 Tue  1    15.6
 Sep           5.0 @ 6h00 Thu 21    22.2 @12h00 Tue 26    13.5
 Oct           2.2 @ 5h00 Mon 30    19.4 @13h00 Sat  7    10.8
 Nov          -0.8 @ 5h00 Mon 27    14.4 @14h00 Sat 11     5.2
 Dec          -4.2 @ 1h00 Sat  9    12.7 @ 9h00 Sat 23     3.8
 All   -6.4 @20h00 Sun  8 Jan  27.7 @12h00 Mon 17 Jul   9.5
 Typical winter week begins Monday 6 Feb,
 Typical spring week begins Monday 17 April,
 Typical summer week begins Monday 3 July.
 Typical autumn week begins Monday 2 October.
 Typical winter week begins Monday 20 November,
*help_end
*item
*name   ALBUQUERQUE NM USA iwec 723650
*aide   ALBUQUERQUE NM USA iwec 723650 was sourced from US DoE web Sep 2005
*dbfl   /usr/esru/esp-r/climate/USA_NM_Albuquerque_iwec
 . . .
```

Figure 16.3 A typical section of a climatelist file.

The list of available climate files

The last ASCII file which is used by ESP-r modules on a regular basis is the so-called *climatelist* file. This file is referenced by the *esprc* file (see above discussion) and includes a list of the climate data sets that were installed on the computer. When the interface of one of the ESP-r modules presents a list of available climate data it scans this file.

Each time you want to add climate data to your computer you should edit this file with a text editor so that the listing will include the new file. There is a detailed discussion of how to use **clm** to add new climate files in Chapter 6. A portion of this file is shown in Figure 16.3.

The *climatelist* file includes the following types of information:

- a display name for the climate data (as seen the the interface list)
- a brief documentation about the climate data
- its location on the computer
- the start and end dates of each of five seasons (winter from 1 Jan, spring, summer, autumn, winter ending 31 Dec). These dates typically were supplied by a person who knows the climate of the region and the social customs of the region.
- the start and end dates of a typical week in each season. There is an facility in the **clm** module which searches for typical weeks based on heating and cooling degree days and solar radiation patterns.
- a block of text up to 60 lines which provides a summary of the climate. This block is auto-generated within **clm** and you can edit it and extend it if required.

17 Version Appendix

In addition to its multi-platform capabilities, ESP-r has three possible styles of interaction on most computing platforms: text-mode interactions, a legacy X11 graphic interface and a newer GTK graphic interface. In most cases the command menu selections are the same. Differences are found in the layout of some types of dialogue, in the file browser facilities, in sensitivity to mouse clicks and keyboard input shortcuts.

First a brief review of the interaction styles. Text mode operation is the most *geeky* on offer. It is primarily used by experts who are commissioning a series of standard assessments. The interface is driven either by user supplied keystrokes or commands included in a script. This mode is also useful for working on a remote compute-server or when only a slow internet connection is available.

The X11 interface has separate regions for graphic feedback, text feedback, menu selections and user dialogues for editing entities and selecting files. Menus items are selected via mouse clicks or by keystroke. This interface is available for all operating systems except for Native Windows. User interactions are uniform across all machine types and each of the ESP-r modules follows the same layout although some take up more room on the screen.

The GTK+ based interface was selected as a replacement for the X11 interface because it can be deployed across a range of operating systems, including Windows. It also has a richer application programming interface and a larger number of in-built features that previously had to be written from scratch in X11.

17.1 Text mode

Text mode operation is available as a command line option for users working on Solaris, Linux, Cygwin and OSX. To use text mode on Native Windows ESP-r must be compiled without graphics. As it is difficult to have two versions of ESP-r on a Windows computer this usually necessitates the use of a second (or virtual) Windows computer in order to have access to wireframe views of models and graphs in the results module.

In Figure 17.1, the project manager has been invoked from a command window (the same command syntax applies in Linux, Solaris, OSX and Cygwin). The list of user options is shown in a double column (options which start with a character or number are activated by typing that character). The prompt is seen at the bottom of the figure.

```
twocv-2:~/testing/cellular_shd/cfg jon$ prj -mode text file cellular
ESP-r Project Manager (prj).
This is a development copy of prj, based on ESP-r release 11.4.

Copyright 2001-2007 Energy Systems Research Unit.

Build information:
  - SVN Source:       prj_dev@r2684 (locally modified)
  - Platform:         Power Macintosh:Darwin:8.11.0
  - Compilers:        gcc-3.3/g++-3.3/g77-3.3
  - Graphics Library: X11
  - XML output:       unsupported


Model Management:
  a introduction to ESP-r
  b database maintenance              .... Import & export ........
  c validation testing               n import CAD file
   .... Model selection ........       export
  d open existing                      archive
  e create new                        .... Model location ........
   .... Current model (none) ....      folders & files
    cfg  :                            .... Miscellaneous ........
    path :                             save model
    root :                             save model as
    title:                            v feedback >> summary
    variants                          * preferences
                                      ? help
  - exit Project Manager

 Model Management:?> []
```

Figure 17.1 An example of invoking the project manager in text mode.

Production work often requires a specific sequence of tasks be carried out. To reinforce this practitioners will often create a script to *drive* ESP-r. For example, during testing a standard set of assessments need to be run and then a standard report should be extracted for each assessment.

Below is a portion of a script called SIMULATE.wc is used to run a standard assessment (with ideal controls active). The script invokes an ESP-r module (bps) with a suitable set of command line parameters and then passes a sequence of keystrokes to the module to control it.

```
#!/bin/csh -fb
set CONFIG=$1
bps -file $CONFIG -mode text <<XXX
c
$CONFIG.wc_res
9 1
15 1
3
1
s
y
ESRU Standard test: $CONFIG
y
y
-
-
XXX
```

Having run the assessment, a second script in the source code validation/benchmark/QA/model/cfg folder named ANALYSE_4 is invoked to start up the results analysis module and cause a sequence of reports to be generated.

```
#!/bin/csh -fb
set RESFILE=$1
res -file $RESFILE -mode text<<XXX
```

```
d # enquire about
>
$RESFILE.data
$RESFILE results
a # summary statistics
h # sensible
a # heating
h
b # cooling
b # temperatures
a # zone db
b
e # zone resultant
b
d # zone control pt
-
-
d # enquire
a # summary statistics
f # zone flux
a # infiltration
f
b # ventilation
m
a # real power
m
b # reactive power
j
b # convective casual gains
-
j
c # radiant casual gains
-
d # solar processes
a # entering from outside
d
c # solar absorbed
i # zone rh
j # casual gains
a # all
-
j
b # convective portion
-
j
c # radiant portion
-
j
e # total occupant gain
-
j
i # total lighting gain
-
j
m # total small power
-
-
-
c # timestep reports
g # perf metrics
j # casual
e # total occupant
-
j
i # total lighting
-
j
m # total small power
-
! # list data
-
-
d
f # energy delivered
g # casual gains distrib
h # zone energy balance
b
b
i # surface energy balance
b
b
* # all surfaces in reception
-
* # all surfaces in office
-
* # all surrace in roof
-
l # monthly
a
b # frequency
b # of temperatures
a # zone db
y
>
-
-
-
XXX
```

Seasoned users of ESP-r often use the Perl language to compose a sequence of assessments and data recovery tasks. Indeed, it is possible to use a script to modify the shape and or composition of a model. Anything that can be done via an interactive session in text model can be included in a script.

17.2 Legacy X11 graphics

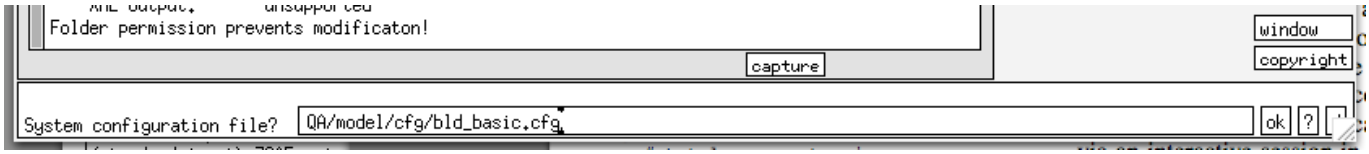


Figure 17.2 Specifying a file name.

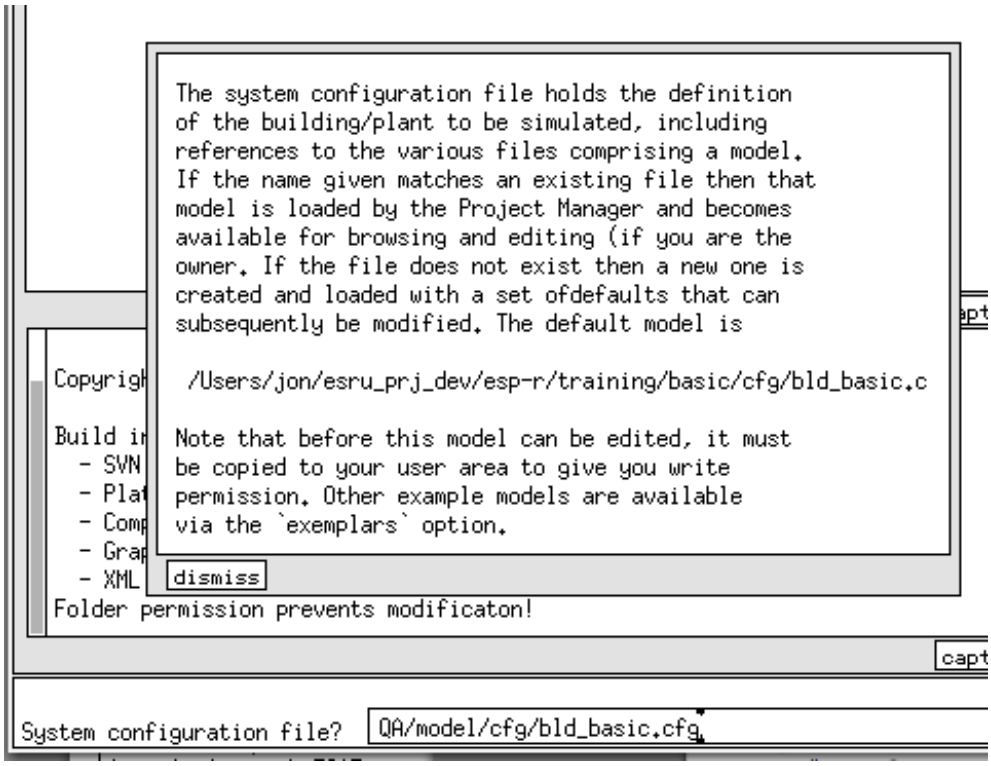


Figure 17.3 Pop-up help for a dialog.

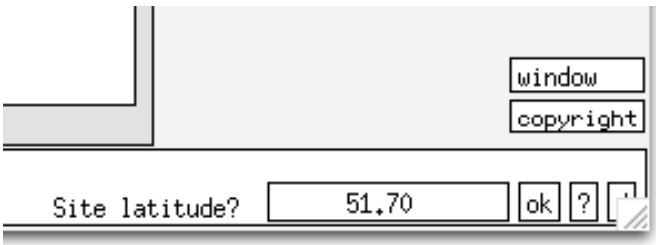


Figure 17.4 X11 real number dialog.

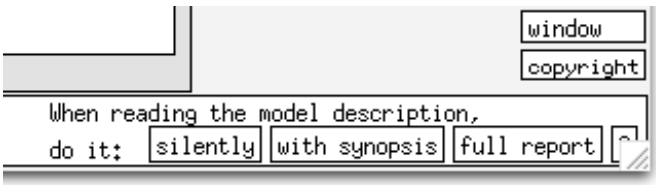


Figure 17.5 X11 radio button dialog.

This style of interaction supports the most complete graphics input functionality. It is also dated in appearance and has limited file browsing facilities. The current plan is to depreciate and eventually remove X11 graphic dependencies as and when the newer GTK library code (see below) are in place.

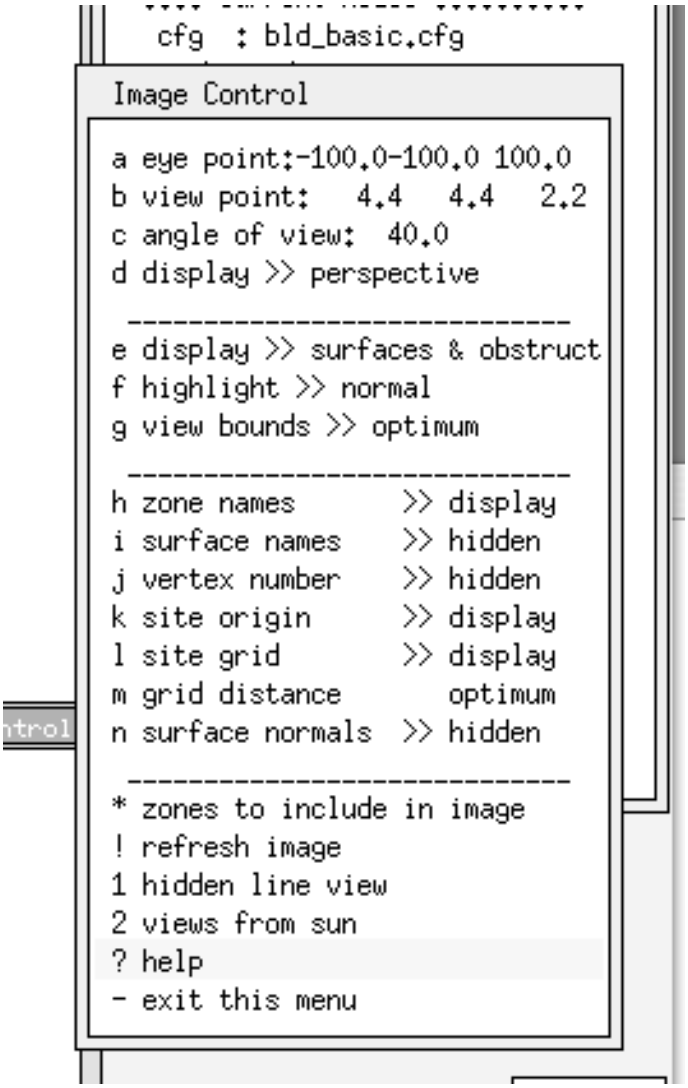


Figure 17.6 X11 wire-frame control menu.

Figure 17.2 is an example of a text dialogue (asking for a file name). The layout is similar to all X11 dialogues - there is a prompt above and/or to the left of the editing box. On the right is an **ok** box, a **?** box and a **default** box. Clicking on the **?** produces a pop-up dialogue (see Figure 17.3). If there are more than about 20 lines of text then up and down arrows will be included to support scrolling. Figure 17.4 is a dialog requesting a real number. Such dialogues usually have range checking included and you might be asked to re-specify the number. Figure 17.5 is the X11 equivalent of a radio button (only one item can be selected). Figure 17.6 shows the X11 control of a wire-frame view and Figure 17.9 is for the GTK interface. In this case a menu dialogue has been used to approximate a pro-forma.

Selection lists where you are able to select more than one entity e.g. Figure 17.7 shows a list of surfaces in a zone that have been selected and the selected items have a * in the right column. There is an `* All items` option which will select the whole list. If you want to remove and item from the selection then select it and the * will be removed.

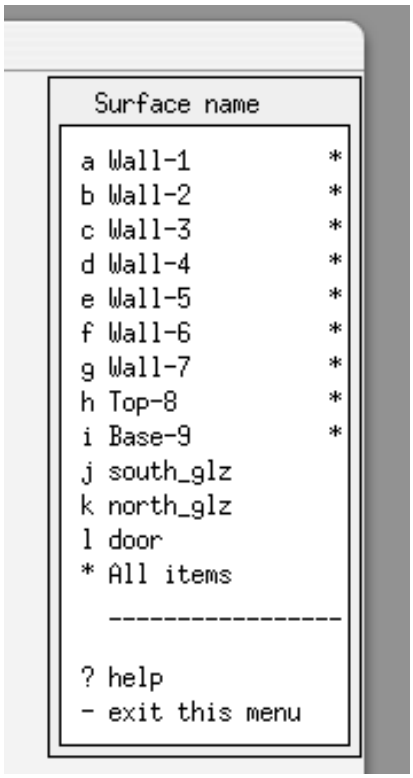


Figure 17.7 X11 entity selection(s) list.

17.3 GTK+ graphics

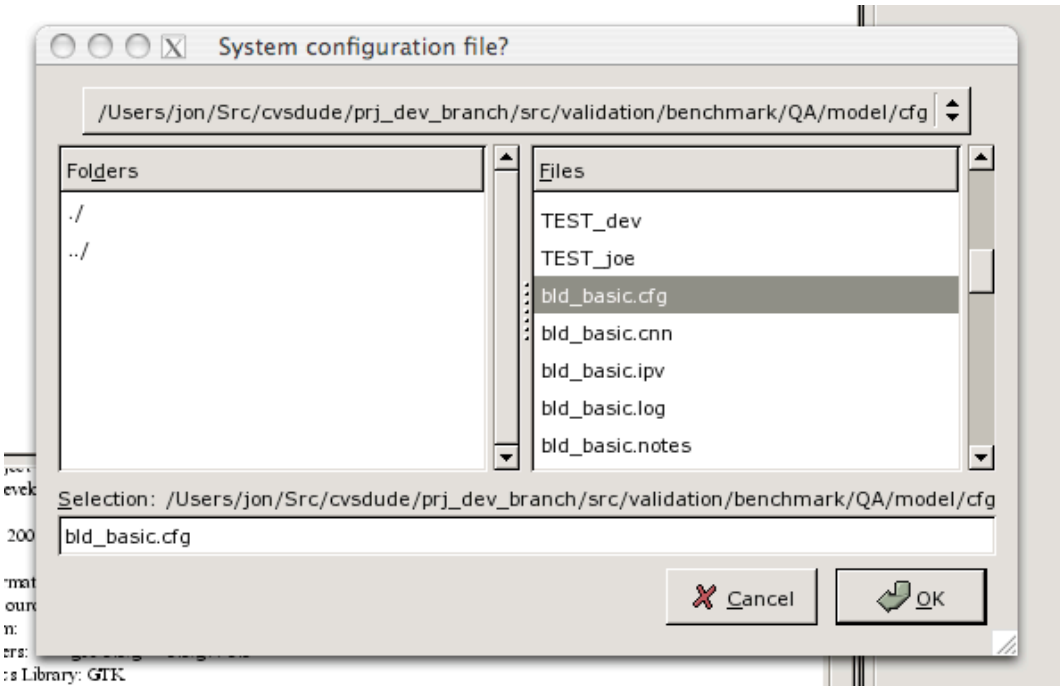


Figure 17.8 GTK file browsing dialog.

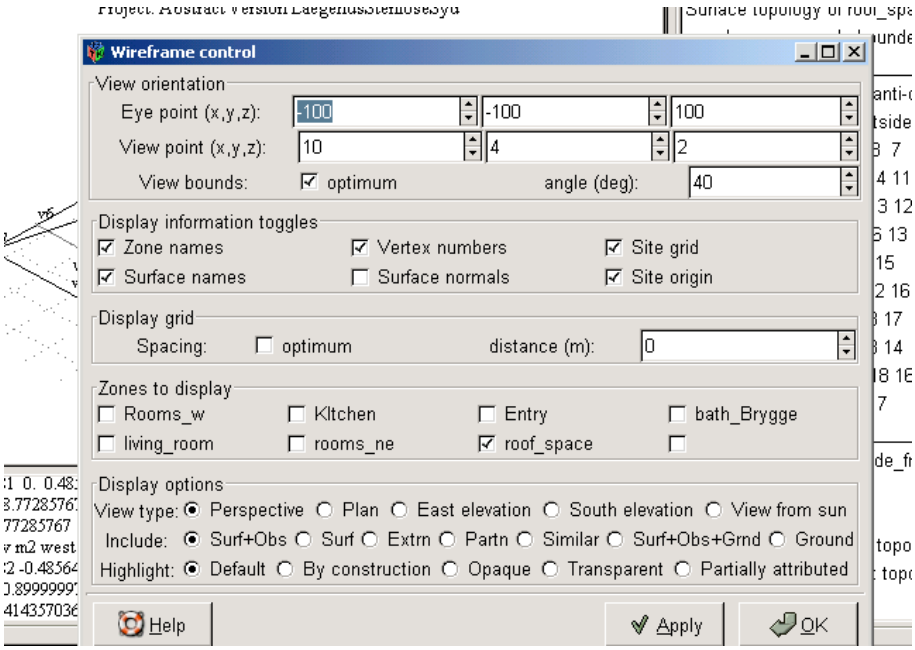


Figure 17.9: GTK wire frame control dialogue.

This style of interaction is more in keeping with what the user community expects. The port is incomplete, however, many users find that they can work around the limitations. The current plan is to continue coding until the full X11 functionality is in place. The development community will then review how the layout of ESP-r might evolve to take advantage of the functionality in the GTK graphics API.

The following Figures are the GTK equivalents to the file specification dialogue, pop-up help message, real

number dialogue and a radio button dialogue. It generally takes fewer lines of code to implement a dialogue in GTK than it does in raw X11.

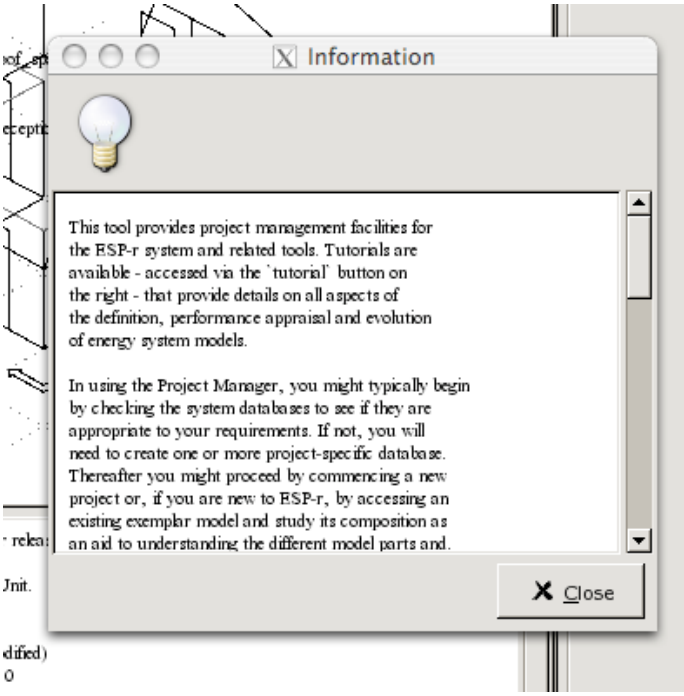


Figure 17.10 GTK popup help.

Figure 17.11 GTK real number dialogue.

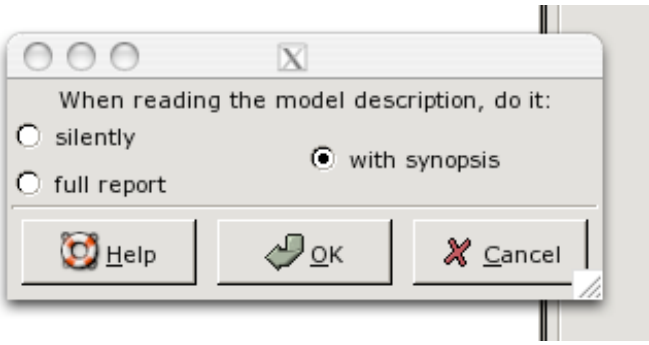


Figure 17.12 GTK radio button dialogue.

ESP-r dialogues normally use English, however GTK does support locale specific text in buttons that are used in some dialogues. Once the X11 code is depreciated it should be possible to consider revising the code structure to support multiple languages if resources could be found.

<< more text here >>

The table below provides a summary of differences. This is followed by specific examples where you might encounter differences.

The chapter is work in progress. More text to be added here.

18 Capabilities

This Chapter is a review of the capabilities of ESP-r for representing the virtual physics of buildings and systems as well as what can be measured and reported. Some of the information is similar to the publication *Contrasting the Capabilities of Building Energy Performance Simulation Programs by Crawley, Hand, Kummert and Griffith of July 2005*.

In that publication the summary of ESP-r is:

> ESP-r is a general purpose, multi-domain (building thermal, inter-zone air flow, intra-zone air movement, HVAC systems and electrical power flow) simulation environment which has been under development for more than 25 years. It follows the pattern of *simulation follows description* where additional technical domain solvers are invoked as the building and systems description evolves. Users have options to increase the geometric, environmental control and operational complexity of models to match the requirements of particular projects. It supports an explicit energy balance in each zone and at each surface and uses message passing between the solvers to support inter-domain interactions. It works with third party tools such as Radiance to support higher resolution assessments as well as interacting with supply and demand matching tools.
>
> ESP-r is distributed as a suite of tools. A *project manager* controls the development of models and requests computational services from other modules in the suite as well as 3rd party tools. Support modules include: climate display and analysis, an integrated (all domain) simulation engines, environmental impacts assessment, 2D-3D conduction grid definitions, shading/insolation calculations, view-factor calculations, short- time-step data definitions, mycotoxin analysis, model conversion (e.g. between CAD and ESP-r) and an interface to the visual simulation suite Radiance.
>
> ESP-r is distributed under a GPL license through a web site which also includes an extensive publications list, example models, cross-referenced source code, tutorials and resources for developers. It runs on almost all computing platforms and under most operating systems.
>
> Although ESP-r has a strong research heritage (e.g. it supports simultaneous building fabric/network mass flow and CFD

domains), it is being used as a consulting tool by architects, engineers, and multidiscipline practices and as the engine for other simulation environments.

18.1 General modelling features

This section provides an overview of how ESP-r approaches the solution of the buildings and systems described in a user's model, the frequency of the solution, the geometric elements which zones can be composed and exchange supported with other CAD and simulations tools.

- ESP-r employs a partitioned solution approach. Simultaneous loads, network air flow, CFD domain, electrical power and system component solution is via custom domain solvers. Interactions between domains are handled with message passing between the solvers at each time-step.
- The building solution is based on matrix partitioning and Gaussian elimination. Partitioning is by thermal zone and zone coupling is handled by message passing at each time-step. This approach conserves memory in the case of large models, but requires resources for message passing.
- Entities in ESP-r are represented as control-volume heat-balance nodes (finite volumes). Such nodes are distributed throughout the fabric and air volumes of the building and within system components. An explicit energy balance is maintained at each zone air volume, at each face of each surface and at each finite volume of used in system components.
- System components support temperature, two component (mosit air and steam) flow, heat injection/extraction and electrical characteristics at each component node.
- Electrical power solution supports mixtures of DC as well as three phase power with real and reactive power. The electrical network can link to casual gains in zones and surfaces which include electrical characteristics as well as system components.
- The mass flow solution works with mixed air and water flow networks of nodes (at zones, system components and at boundary locations) as well as a range of flow components. Control logic can be applied to flow components to approximate mechanical or user interventions. The solver is highly optimised and can solve networks of hundreds of nodes with minimal computational overhead. It typically runs at the same frequency as the zone solver with the option to iterate (useful for models with large openings).
- Iteration supported in system component and network flow solution domains. Zone solver can be adjusted from fully implicit to fully explicit but defaults a Crank-Nicolson formulation. Conduction defaults to 1D and models can include 2D and 3D conduction (if additional data is supplied).
- Domain interactions supported e.g. mass flow solution feeds system component and zone solution at each time-step.

- Zones can be assessed with a mixture of environmental controls, including free float and user undersized controls as well as supporting radiant gains and injection of heat into layers of a construction.
- Simulation frequency is one minute to one hour for zone and air flow domains. System and electrical components solved from one second to one hour. Time-step controller with support for rewind to start of day for exploring optimal start regimes.
- Zone geometry is based on 3D polygons of arbitrary complexity (including explicit representation of internal mass). Shading devices are block shapes. There are a number of geometric rules: a) zones must be fully bounded, b) surfaces must be flat, c) edge ordering defines outward face of surface, d) unbounded edges detected, e) internal mass requires back-to-back surfaces, f) zones can be embedded within other zones. All zone surfaces take part in the zone energy balance.
- Glazing is a surface with optical property attributes which supports solar transmission and absorption at each layer in addition to the convective, conductive and radiant exchanges of opaque surfaces. Optical properties can be subject to control action. Frames can be represented explicitly or via adapting the properties of the surface representing glazing.
- Surfaces can have attributes for phase change materials, temperature dependant conductivity, electrical characteristics (e.g. integrated PV), moisture adsorption, contaminate emission properties.
- Facility to import DXF files (V12) which confirm to specific standards of layer naming and use of 3D entities.
- Facility to export DXF files (V12), EnergyPlus (geometry, constructions, casual gains), VRML worlds, Radiance visual simulation models.
- Measured data (one minute to one hour frequency) of temperatures, setpoints, weather data, casual gains can be associated with a model.
- Ideal environmental control systems can be defined (in addition to component based system descriptions).

18.2 Zone Loads

This section provides an overview of ESP-r's support for solving the thermophysical state of rooms: the heat balance underlying the calculations, how conduction and convection within rooms is solved, and how thermal comfort is assessed.

- An explicit energy balance is maintained at each zone air volume and at each face of each surface. The default treatment is to use three finite volumes for each layer of each construction. Typically, materials over about 200mm thick are subdivided into multiple layers to increase the efficiency of the solution as well as providing additional points for recording temperature.
- The minimum size of a zone is ~1cm^3. Rooms with dimensions greater than 100m probably should be subdivided. Minimal surface

dimensions are ~1mm and there is no specific maximum size. Surfaces can include ~24 edges but complex surfaces or surfaces with large dimensions may not be well represented by 1D conduction. Minimum thickness of a construction layer is ~0.2mm although care should be taken when thin and thick layers are adjacent.

- A number of boundary condition types are supported in ESP-r: a) exterior, b) other side has similar temperature and radiation, c) other side has fixed temperature and radiation, d) other side is a surface in this or another zone, e) standard or user defined monthly ground temperature profile or a 3D ground model, f) adiabatic (no flux crosses), g) other side is part of a BASESIMP foundation description, h) CEN 13791 partition.
- Air gaps are typically treated a resistance layers which are sensitive to surface orientation. Glazing using alternative gasses or with coatings must be approximated by altering the air gap resistance. Air gaps can be explicitly represented as thermal zones and optionally included in an air flow network (such explicit treatments do not scale well but do support explicit treatments of radiation and convection across air gaps).
- There are ~20 internal convection regimes and 25 external convection correlation's in addition to user defined heat transfer coefficients. Conditions at inside and outside face are evaluated at each time-step. Some outside regimes are sensitive to the angle of incidence of the wind as well as whether the surface is on the windward or leeward side. Heat transfer coefficients derived via a CFD solution can be applied at the next time-step.
- Internal mass can be explicitly represented and these surfaces take part in the full energy balance as well as solar insolation patterns and long wave radiant exchanges.
- Radiation view factors can be calculated for zones of arbitrary complexity and shape, including internal mass.
- Fanger thermal comfort as well as MRT and resultant temperature reported. If sensor bodies are defined then radiant asymmetry is also reported.

18.3 Building envelope and daylighting

This sections is an overview of the treatment of solar radiation outside a building as well as its distribution within and between zones. Outside surface conduction is also discussed.

- The default treatment is to assume no shading and that the distribution of insolation is diffusely distributed within a room. If there are external sources of shading these are represented as opaque non-reflective block shapes. Shading calculations (direct and diffuse) are done for each hour on a typical day of each month. Diffuse shading can be based on isotropic or anistropic sky conditions.
- Insolation distribution (including radiation falling on internal mass surfaces) can also be predicted at each hour on a typical day of each

month.

- Beam solar radiation is tracked to first absorption and then diffusely distributed. Solar radiation passing into adjacent zones is treated as a diffuse source. Solar radiation passing out through a facade is assessed.
- Optical data on transmission and absorption at five angles is typically used and can be imported from Window 5.1 or 5.2. Data from WIS requires additional editing. Bidirectional optical properties and control is supported for those with experimental data.
- Seasonal adaptation of shading devices requires separate models, each pointing to obstruction descriptions for that season.
- Optical properties can be switched based on a range of criteria (the number of layers is required to be constant). There is a facility to substitute an alternative construction as well as alternative optical properties. The requirement for a constant number of layers poses a challenge for the representation of movable blinds and shutters.
- Daylighting control can be based on split-flux method, user defined daylight factors or time-step use of the Radiance visual simulation suite to compute lux on a sensor. Multiple circuits can be treated in each thermal zone.
- Some users choose to explicitly represent opaque blinds as sets of surfaces. In the case of blinds between glass this is often approached by treating the blind as a layer with the construction which has optical properties approximating the transmission through the slats and openings. If the optical properties of this layer are switched then the absorption characteristics of the blind layer change (as does the thermal state of the construction).
- Conduction is typically represented in 1D but can be switched to 2D and 3D (the input data requirements increase considerably).

18.4 Infiltration ventilation and multi-zone air flow

This section provides an overview of how air movement, either from the outside or between rooms or in conjunction with environmental systems is treated.

- The simple approach to air movement in ESP-r is to create schedules of infiltration (air movement from the outside either natural or forced) and ventilation (air movement between zones). Control can be applied to these scheduled flows based on a range of criteria e.g. increase infiltration to 2ach if room goes over 24 degrees C.
- The intermediate approach to air movement is to create a network of flow nodes and components and solve the leakage distribution at each time-step based on the current boundary driving forces as well as stack and pressure distributions inside. Most of the underlying component representation are derived from the literature. In common with most other mass flow solution approaches there are some gaps in

the provision of component types.

- Control can be applied to mass flow components to approximate occupant interactions or mechanical flow controls. Combinations of controls in sequence and in parallel are used to create complex regimes.
- The highest resolution approach is to define a computation fluid dynamics domain with one zone of the model (optionally in addition to a flow network). The CFD domain is in a rectangular co-ordinate system with optional blockages. Solution is typically transient 2D or 3D. There are a range of wall functions available as well as equation types.
- The CFD solution supports one way and two way conflation. The latter takes its boundary conditions from the zone solution and returns heat transfer coefficients. There is an adaptive controller which re-forms the CFD domain at each time-step based on changes in boundary conditions and the flow patterns predicted from an initial course CFD assessment.
- The CFD solution can also use flow boundary conditions from an associated mass flow network. This allows wind pressures and flows with other zones to be assessed at each time-step. Iteration is used to negotiate between the mass flow and CFD domain solvers.
- The CFD solution can include cells which are heat and contaminate sources. The former can be associated with zone casual gain schedules so that there is a temporal aspect to such gains.
- Zone air volumes are assumed to be well mixed at one temperature. Stratification requires either the use of a CFD domain or subdivision of physical spaces into multiple thermal zones with a mass flow network used to assess air exchange.
- A mass flow network can include elements associated with natural ventilation and buoyancy driven flow as well as components within a mechanical system. Thus it is possible to create an air based solar collector from a collection of explicit zones and a flow network (as an alternative to a component based approach).
- For models of high resolution there is a post-processor which determines if specific species of micotoxin will grow on surfaces of a model.
- Architectural elements such as Trombe-Michelle walls and double skin facades are usually composed of multiple zones with a mass flow network to support air movement assessments.
- ESP-r includes a database of wind pressure coefficients (at 22.5 degree intervals) which can be associated with wind pressure boundary nodes in a flow network. Some users populate this database with data from wind tunnel tests or CFD runs.

18.5 Renewable energy systems and electrical systems

This section is an overview of options for representing renewable energy systems either as components within an ESP-r model or via 3rd party tools. Including an electrical power network

in a model allows a number of issues related to renewable energy systems to be addressed.

- Some system components represent components of renewable energy systems such as generators, fuel cells and batteries. These can be tested independent of or switched into the power grid. Mixtures of 1/2/3 phase AC and DC can be described.
- The electrical solvers works at the same time-step as the system components and yields real and reactive power flows, power losses, current magnitudes and phase, voltage magnitudes and phase, phase loadings.
- ESP-r data can be exported for use with supply and demand matching tools such as MERIT (also available from ESRU).

18.6 Ideal environmental controls

This section is an overview of ESP-r's approach to ideal (from a control engineers perspective) zone controls.

- For early stage design issues ESP-r users tend to use ideal zone controls to represent environmental controls as loops of sensors and actuators with a range of control laws.
- Ideal zone controls can be combined with control of mass flow components to increase the resolution of models. Mass flow components can be used to represent some aspects of duct work in mechanical systems (ESP-r has not been designed to be used as a duct design tool).
- Sensors in ESP-r can be located at a number of positions within the building (the zone air volume, at or within a surface, at a flow node or outside of the building for use with climate variables). Actuators in ESP-r can be located at the air node, at or within a surface, at a flow component etc.

Zone control loops include a schedule of control laws are applied as required to approximate many environmental control regimes. The zone controls include:

- basic ideal control with maximum and minimum heating and cooling capacity, heating setpoint, cooling setpoint and moisture injection.
- free floating control
- fixed injection/extraction with heating injection, cooling extraction, heating set-point and cooling set-point.
- basic proportional control with maximum and minimum heating injection, maximum and minimum cooling extraction, heating set-point, cooling set-point. There is a throttling range and optional integral action time and/or derivative action time.
- A multi-stage controller with hysteresis. There are three stages of heating and three stages of cooling. There are heating off and cooling off set-points as well as dead bands for heating and cooling and heating and cooling set-points.
- Variable supply controller with or without available cooling. It includes a maximum supply temperature, minimum supply temperature, air flow rate room heating and cooling

set-points.

- Separate ON/OFF controller which includes heating and cooling capacity, heating on below, heating off above, cooling on above and cooling off below set-points.
- Ideal match temperature controller which is given a maximum heating and cooling capacity and a list of sensors and their weightings as well as scaling factors. A typical use is to control a boundary zone to a measured temperature or to match the temperature in another zone. There is also an ON/OFF controller which can be used to match measurements or other zone temperatures.
- Time proportioning control which includes heating and cooing capacity, heat on and off set-points, cooling on and off set-points, minimum on times and off times. Useful for equipment that has a slow cycle time.
- Optimal start logic with heating capacity, desired setpoint, desired time of arrival, minimum time difference with optional start time. There is also an optimal stop controller.
- Slave capacity controller points to a common sensor and forces the zone actuator to act as the master actuator but with a user defined heating and cooling capacity.
- VAV approximation controller which includes a reheat capacity, supply temperature, room set-point, maximum and minimum flow rate

18.7 Component based systems

This section is an overview of ESP-r's component-based approach to describing environmental systems. There are a number of component types (some are listed below) which can be linked together to form a range of environmental control systems. Some devices are represented by several components - for example there is a single node radiator as well as an eight node radiator - so the user has a choice of component resolution.

As with zone controls, system components can be included in system control loops. There are a number of control laws available depending on whether flux or flow is to be controlled.

- Air conditioning steam/spray humidifier, water/steam flow multiplier, water/steam flow converger and diverger
- Air conditioning cooling coils with flux control, a counterflow cooling coil with water mass flow rate control, a counterflow cooling coil fed by WCH system, a two node cooling coil with specific fin and tube details.
- Air conditioning heating coils with flux control, a counterflow heating coil with water mass flow rate control, a counterflow heating coil fed by WCH system
- Plate heat exchanger, air/air exchanger, heat exchanger segment, duct, duct damper
- Cooling tower
- Centrifugal fan

- Wet central heating boilers: non-condensing boiler, with on/off control, with aquastat control, calorifier, modulating boiler, TRNSYS type 6 WCH boiler storage water heater
- Wet central heating radiators: basic (one node), exponent model radiator, WCH thermostatic radiator valve, mechanical room thermostat.
- WCH pipes: pipe, converging 2-leg junction, converging multi-leg junction, heat transfer tube, heat transfer tube with transport delay, insulated pipe with transport delay, flow control valve
- WCH basic chiller or heat pump, water cooler, compressed gas cylinder
- WCH generic liquid/liquid heat exchanger and gas/liquid heat exchanger and water/air heat rejector
- Oil filled electric panel radiator
- CHP engine components: one node and three node representations.
- Hydrogen appliance (generic generator supplied by hydrogen source)

18.8 Environmental emissions

The emissions associated with the energy use of buildings can be tracked via user supplied conversion factors for heating and cooling and small power loads as well as loads which are not attributable to a particular zone.

There is a rarely used facility that allows users to define environmental impacts associated with the assembly and transport and disposal of building components.

18.9 Climate data

This section gives a summary of how ESP-r uses climate data and how users access and manipulate this data.

ESP-r holds the following climate data in both ASCII and binary file format. Solar data can be in two forms. The normal file formats support hourly data.

- Diffuse solar on the horizontal (W/m**2)
- External dry bulb temperature (Deg.C)
- Direct normal solar intensity or global horizontal (W/m**2)
- Prevailing wind speed (m/s)
- Wind direction (degrees clockwise deg from north)
- Relative humidity (Percent)
- Site description, latitude and longitude difference from the local time meridian.

There is a conversion utility which is able to read EnergyPlus EPW weather data and extract the required data fields. ESP-r also works with sub-hourly weather data via a so-called temporal file.

There is an facility that scans a climate data set to determine best fit weeks in each season based on heating and cooling degree days and radiation levels. It also reports initial scaling factors that can be used to convert from short period assessments to seasonal performance data.

18.10 Results reporting

This section is an overview of how building and systems performance data is accessed in ESP-r. The standard approach differs from many other simulation suites in that one or more custom binary databases are created depending on the number of domain solvers used in a particular model.

- The zone solver writes out a number of items at each time step which related to temperatures of the zone air and surfaces as well as fluxes associated with a zone air energy balance and surface energy balances.
- The plant component solver writes out the temperature, 1st phase and 2nd phase flows at each node of each component.
- The mass flow solver records the pressure and temperature at each node, pressure difference across each network connection as well as stack pressure and mass flows along each connection.
- The electrical power solver writes out the real and reactive power, power loss, current magnitude and phase, voltage magnitude and phase, phase loading for each node in the electrical network.

ESP-r includes a results analysis module which is able to read these binary results files and, for any of the stored variables report on the data in various formats and undertake statistical analysis.

- Graphs of time vs variable with option of multiple vertical axis and support for combinations of data.
- Graphs of variable vs variable to look for trends and correlation's between data types
- Statistics with maximum value and time of occurrence, minimal value and time of occurrence, mean and standard deviation
- Statistics with number of hours over and under a specified value
- Time step listings in multiple columns with various separators as required by third party applications
- Frequency bins and cumulative frequency bins in tables and graphs.
- Zone and surface energy balances as well as individual reporting of all contributor variables in the energy balances.
- Energy demand over time including hours of use.
- Comfort in terms of PMV, PPD, radiant asymmetry, resultant temperature, MRT.
- Monthly gains and losses for a number of variables in each zone.

There is an facility which generates XML files based on a description of variables to save during a simulation. The list of items to capture is typically generated by editing a separate specification file.

There is a facility which allows users to include a range of performance metrics for a model called an Integrated Performance View. This description is used by the results analysis module to generate a report based on information in one or more simulation files.

18.11 Validation

Testing of software has been an ongoing activity for the ESP-r development community and takes several forms: assuring the quality of the code in ESP-r, testing that the underlying computations are correct and detecting differences with the predictions of other tools. ESP-r has been involved in the following projects:

- IEA Annex 1 (1977-1980). Inter-program comparison of 19 different tools
- IEA Annex 4 (1979-1982). Inter-program comparison of a commercial office building with nine tools over four years.
- IEA Annex 10 (1984-1986). Inter-program comparison of HVAC systems.
- IEA Annex 21 (1988-1993). Inter-program and analytical comparison commonly known as BESTEST.
- SERC validation project (1988). Inter-program comparison undertaken by several Universities and research groups in the UK with extensive analytical tests.
- UK Energy Technology Support Unit Applicability study was carried out over seven years and focused on passive solar houses.
- EC PASSYS project (1986-1993) combined outdoor test facilities (in 14 locations) with model predictions.
- British Research Establishment/ Electricity de France empirical validation study of a BRE office building and a BRE house.
- BESTEST, RADTEST, Home Energy Rating System BESTEST were carried out at various times and by various teams and focused on radiant heating systems (RADTEST) and air conditioning systems and furnace models (HERS).
- CEN 13791 standard required a number of extensions to ESP-r to accommodate the unusual physics assumed in the standard.