

A PROPOSED METHOD FOR GENERATING,STORING AND MANAGING LARGE AMOUNTS OF MODELLING DATA USING SCRIPTS AND ON-LINE DATABASES

Spyros Stravoravdis¹, Dr. Andrew Marsh²

¹Welsh School of Architecture, Cardiff University,
Cardiff, UK. Email: spyros@cardiff.ac.uk

²Welsh School of Architecture, Cardiff University,
Cardiff, UK. Email: marsha@cardiff.ac.uk

ABSTRACT

With computational analysis and simulation becoming an increasingly important part of the building design process, the complexities of dealing with the huge amount of data that this can produce can be overwhelming for many designers. This is especially true in large projects where several different types of analysis may be required, spread out over various different areas of the building and dealing with many different parameters. Without a mechanism for managing and interrogating all the output, important trends and relationships within the data can easily be missed.

This paper argues that the process of making sense of all this output in such a way that it is possible to make informed design choices is just as important as performing the analysis itself. As a result, it proposes an innovative means of generating and storing output data using scripts to automate parametric calculations and an on-line database for storing this data for access by any member of the design team.

A case study of lighting and shading in a high-rise complex urban environment is presented where the use of such a generation and data manipulation system is demonstrated.

INTRODUCTION

For the building shown in Figure 1, located in a complex high-rise urban environment in Hong Kong, a study was undertaken to reveal the effects on daylight availability on each flat. This study took into consideration their exposure direction, height, separation distance from other buildings, colour of facades, glazing systems and shadowing effects of the building itself as well as the adjacent buildings.

The building in position G is a 40 storey residential complex, for which the daylight availability on each window is dominated by exposure direction and height. The building's height (120m) and crossed plan introduces plenty of self-shading which, coupled with the overshadowing of equally high nearby buildings, produces a complex urban environment where parameters such as daylight factors can vary significantly among nearby flats.

This creates a necessity to calculate such values for each flat over the whole height of the building. Simplifying the geometry of the building, or producing simplified models of the building and site, would not allow for all the complex effects present in the site to be properly accounted for, thus producing less accurate results.

In order to assess all those parameters, shading masks for each window, internal daylight factors for each flat, vertical (external) daylight factors for each window and incident solar radiation on each wall had to be calculated. With all this information available, a data analysis would reveal and quantify the effects of each of the above parameters.

To generate all the necessary data, each window and room on every flat for each floor would have to be modelled and simulated. There are 4 distinct wings in this building (North, East, West, South), with the North and South being identical in plan, which is also true for the West and East wings (Figure 2). Each wing comprises of 5 flats with the North and South wings having 6 rooms and 10 windows per wing and the East and West wings having 8 rooms and 12 windows per wing (excluding kitchens and bathrooms). This would require the modelling and simulation of performance of 28 rooms and 44 windows per floor, which would amount to 1120 rooms and 1760 windows over 40 floors.

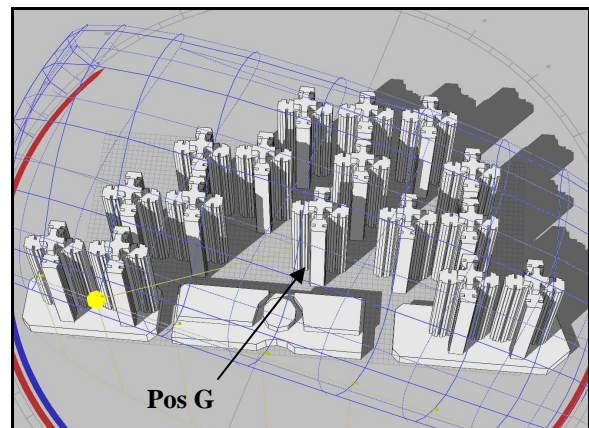


Figure 1 Shadow diagram and annual Sun-Path for a high-rise residential estate in Hong Kong

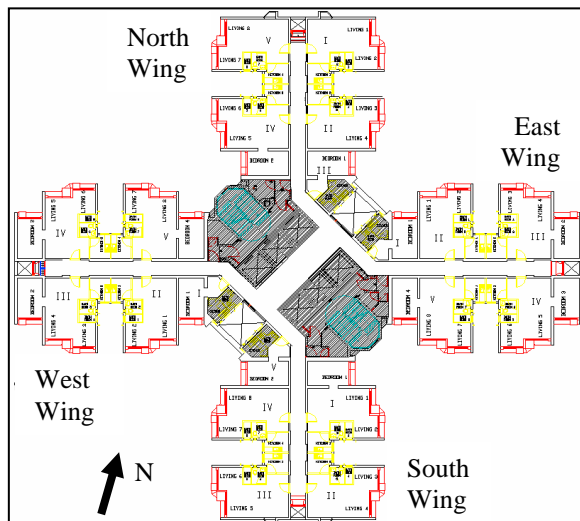


Figure 2 Layout of a typical floor

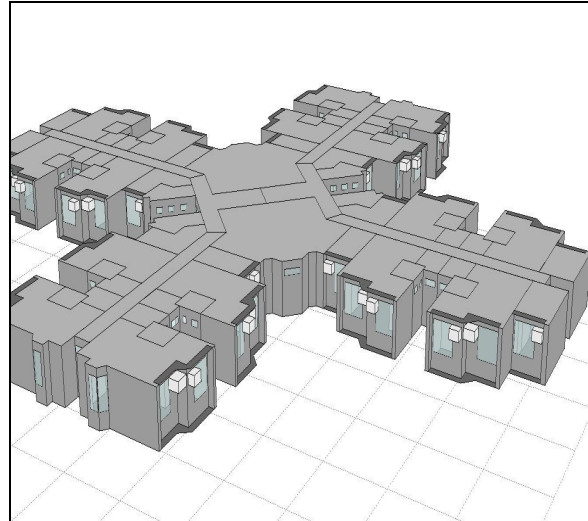


Figure 3 Initial base model of the ground floor flats, from which all other models were derived

DATA GENERATION

To generate the necessary data for this parametric study, the modelling of 28 rooms and 40 windows was required for each of the 40 floors. For each room an average of 200 sample points were taken to show the daylight factor variation within the room. For each window a vertical (external) daylight factor was calculated, as well as the incident solar radiation on its surface. The internal daylight factors were calculated again, to show the effect of different glazing systems (single, double) and of different external reflectances (0.1, 0.3, 0.5, 0.7, 0.9) from adjacent buildings. Overall, about 280 models were required, producing more than 2,500,000 data points.

For the modelling of this parametric study, a combination of three different software was used; ECOTECT (Marsh A., 1996) for its ease of use in constructing models and visualizing results; RADIANCE (Ward G., 1994) for its accuracy in daylighting studies; and LUA-scripting (Ierusalimsky et al., 1996) for controlling and manipulating the models and their parameters within ECOTECT.

LUA-scripting has been incorporated into ECOTECT allowing control and manipulation of any model from a script. Therefore, with a few lines of code, it is possible to load an ECOTECT model, manipulate its geometry, alter certain parameters, export to a variety of other dedicated simulation software, run the calculations there and import the analysis results back into the ECOTECT model.

The method used for this study, makes use of this capability by controlling the whole modelling and simulation process through scripts. There are two basic steps to this method:

1. Construct an initial base model from which all the other necessary models will be derived (i.e. for the building in Figure 1, all the flats on the ground floor were modelled in ECOTECT as shown in Figure 3).
2. Compile LUA-Scripting code to control the remaining modelling and simulation process:
 - a. Derive all other necessary models by updating geometry and parameters (i.e. another 280 models were derived from the initial ECOTECT model, by updating the floor height, glazing system and external reflectance). A sample of the code is shown in Figure 5.
 - b. Export the models to other software and run the appropriate calculation there (all models were exported to RADIANCE)
 - c. Import the simulation results back to the relevant model (results obtained from RADIANCE where imported back to the ECOTECT models)
 - d. Export all simulation results on a file as individual records. Each data point calculated forms an individual record.

Project	Model	Date	Time	X-Position (mm)	Y-Position (mm)	Z-Position (mm)	Wing	Floor
P100213	First.eco	12/12/2004	12:07	232197	292029	6700	C	1
P100213	First.eco	12/12/2004	12:07	232969	294609	6700	C	1
P100213	First.eco	12/12/2004	12:07	232967	299319	6700	C	1
P100213	First.eco	12/12/2004	12:07	232189	301729	6700	C	1
P100213	First.eco	12/12/2004	12:07	230335	304363	6700	C	1
P100213	First.eco	12/12/2004	12:07	234209	310349	6700	B	1
P100213	First.eco	12/12/2004	12:07	236368	308486	6700	B	1
P100213	First.eco	12/12/2004	12:07	238949	307719	6700	B	1
P100213	First.eco	12/12/2004	12:07	243658	307722	6700	B	1
P100213	First.eco	12/12/2004	12:07	246546	310012	6700	B	1
P100213	First.eco	12/12/2004	12:07	236543	310213	6700	B	1
P100213	First.eco	12/12/2004	12:07	237652	311021	6700	A	1
P100213	First.eco	12/12/2004	12:07	286527	320022	6700	A	1
P100213	First.eco	12/12/2004	12:07	275643	321021	6700	A	1
P100213	First.eco	12/12/2004	12:07	296356	321145	6700	A	1
P100213	First.eco	12/12/2004	12:07	303829	323859	6700	D	1
P100213	First.eco	12/12/2004	12:07	313591	326136	6700	D	1

Figure 4 Extract of simulation data stored as individual records on a text file

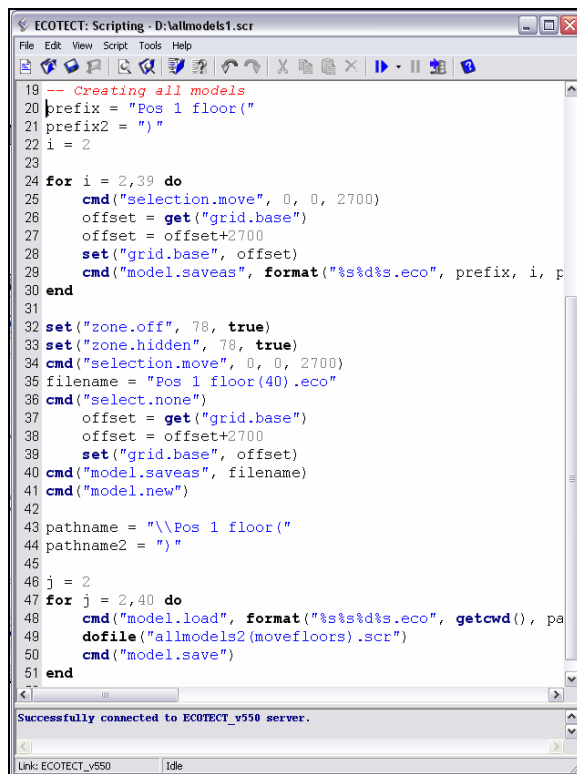


Figure 5 Extract from a Script used in the generation of other ECOTECT models

In order to construct the initial core model, from which all other models were derived, careful consideration was given to the geometry drawn as well as the parameters modelled. For this example, the geometry modelled included all flats on the ground floor, incorporating details of the façade of the building, such as the exposed concrete floors and the attached AC units on the walls. Parameters, such as glazing type and external reflectance, were set to single glazing and 0.5 respectively, for the walls of the building itself and of the adjacent buildings.

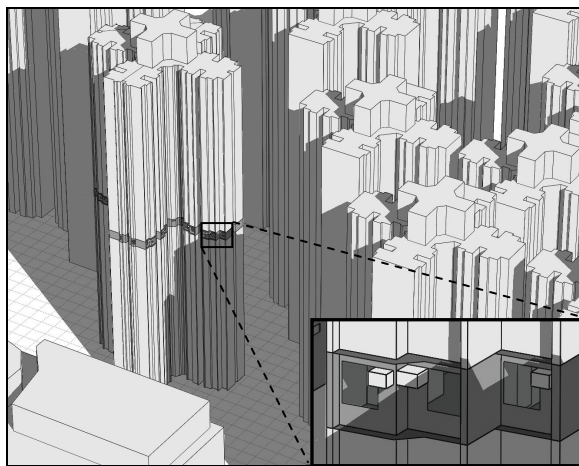


Figure 6 Model of flats on the 20th floor (highlighted in dark grey half way through the building)

To generate all the other models, a simple script code was compiled whereby the previously constructed model was invoked to open in ECOTECT, move all of the ground floor flats up by 3m and save that model under a different name. Where necessary, parameters like glazing type and external reflectance were updated and saved under different model names. An example of a model created for the middle floor can be seen on Figure 6.

Once all the models were created, further scripting was necessary to export them into a RADIANCE compatible format. All the models, were exported as individual files so that RADIANCE calculations could run in parallel on a cluster, making use of its parallel processing capability (Wilkinson & Allen, 2004) and thus reducing CPU processing time significantly. If parallel processing is not an option, the capability also exists to export all models into one single file, so that RADIANCE calculations could run uninterrupted without having to run each file individually, thus saving on monitoring time. When the RADIANCE simulation results had finished, another script imported the data back into the relevant ECOTECT model.

The simulation of so many parameters produced a huge amount data. It became clear that visualisation and interpretation of simulation data by closely observing each ECOTECT model separately (as shown on Figure 6) would be a tedious and time consuming process. Exporting the data to a commonly used program like EXCEL would not be possible, since the data produced was in the range of millions. A data management system was required, where vast amounts of data could be easily stored, accessed and manipulated. The solution was found in the form of an on-line database system, where the simulation data exported by a script could be easily stored.

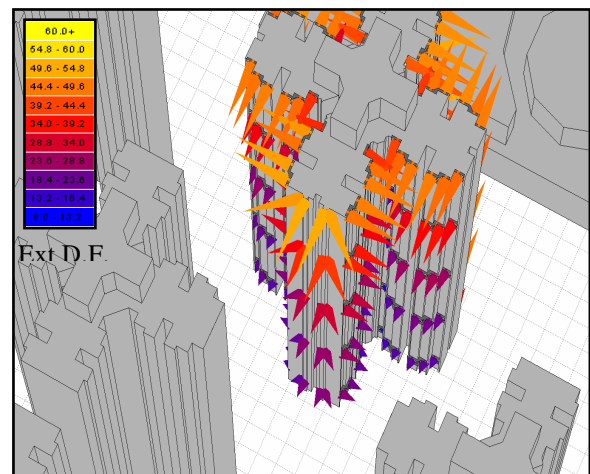


Figure 7 Data visualisation of vertical daylight factors on windows at 5 different levels

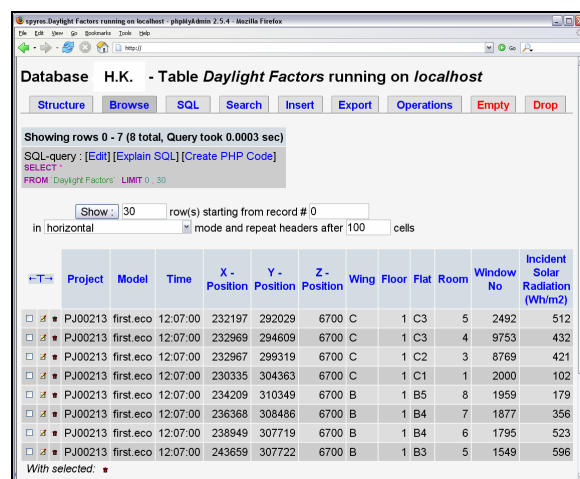
DATA MANAGEMENT

Analysis data was stored on a server with the use of an online database system. MySQL ('My' Structured Query Language) is a freely available open-source database management system, which allows fast access and analysis of huge amounts of data.

Data was imported into this database by exporting the analysis results as individual records, in this case stored as SQL INSERT statements. For every data obtained by the simulation, whether that is a value for a point in space or an average value for a whole surface, a unique record is produced. Each record is stored into a table format with the simulation data point being the variable and all other columns of the table being the parameters describing that variable. That record, in addition to containing the variable calculated (i.e. daylight factor, Incident Solar Radiation), must also contain as many parameters as is necessary to describe its position in space and the relevant group it may belong to.

Since each variable (i.e. daylight factor) is treated as a record, it is necessary to embed enough information on each record so that useful analysis can be conducted later. For example, it is necessary to associate data with building elements, either as rooms, floors or even individual windows and objects so that a group of data, such as West facing windows between floors 10-20, or North facing rooms between floor 35-40 could be easily identified for further analysis. Additional information on each record can include the project code of the study, the person conducting the simulation, the date simulations were completed, the model filename, among others. An example table containing such records in MySQL is shown in Figure 8.

The more parameters a record contains, the easier it would be for groups of data to be identified and future data analysis to take place.



The screenshot shows a web browser window displaying a MySQL database interface. The title bar indicates 'MySQ: Daylight Factors running on localhost - phpMyAdmin 2.5.4 - Mozilla Firefox'. The main content area shows a table named 'Daylight Factors running on localhost'. The table has columns: Project, Model, Time, X-Position, Y-Position, Z-Position, Wing, Floor, Flat, Room, Window No, and Incident Solar Radiation (Wh/m2). The table contains 10 rows of data, all with a time of 12:07:00. The interface includes a search bar, a SQL query editor, and a table view toggle.

Project	Model	Time	X-Position	Y-Position	Z-Position	Wing	Floor	Flat	Room	Window No	Incident Solar Radiation (Wh/m2)
PJ00213	first.eco	12:07:00	232197	292029	6700	C	1	C3	5	2492	512
PJ00213	first.eco	12:07:00	232969	294609	6700	C	1	C3	4	9753	432
PJ00213	first.eco	12:07:00	232967	299319	6700	C	1	C2	3	8769	421
PJ00213	first.eco	12:07:00	230335	304363	6700	C	1	C1	1	2000	102
PJ00213	first.eco	12:07:00	234209	310349	6700	B	1	B5	8	1959	179
PJ00213	first.eco	12:07:00	236368	308486	6700	B	1	B4	7	1877	356
PJ00213	first.eco	12:07:00	238949	307719	6700	B	1	B4	6	1795	523
PJ00213	first.eco	12:07:00	243659	307722	6700	B	1	B3	5	1549	596

Figure 8 Example of a MySQL page containing records of simulation data

Once the simulation data is imported back on each ECOTECT model, a script can easily export all the results in any desired table format so that they can be directly uploaded to the MySQL database.

DATA ANALYSIS

The use of scripting and online databases, allowed us to generate all the required simulation data and store them at a convenient location, while still being able to access and manipulate them fast. With MySQL, a part of the data can be easily invoked and presented on a table format. A range of calculations can then be performed on all or part of the data, by writing a simple code and the results presented again on a table format. An example would be to invoke a table containing the average internal daylight factor for each flat on the East facade of the building for all 40 storeys. In this case, the average of 200 data points per room would have to be calculated and then a table produced for each flat on the East facade and all that repeated for each floor. Such a table could be produced within seconds, from a database larger than two million records. The table, can then be exported to an EXCEL sheet to produce graphs.

Even though such a procedure can be helpful in analysing data, it requires the knowledge of SQL code. In addition to that, data in MySQL can only be presented as tables and analysing data on tables can still be tedious and time consuming, especially for such large amounts of data. A solution was therefore required that would allow fast access of the MySQL database and provide a more userfriendly interface with a graphical representation of data. This was found in the form of HTML code with PHP.

By designing an HTML webpage, the user can define the needs of a project and provide a user friendly interface to work with. The PHP (Hypertext Preprocessor) is the underlying code of the HTML page, that allows access to the MySQL database. PHP, is a freely available open-source scripting language, embedded into HTML and used to create dynamic web pages.

The possibilities of using MySQL databases and PHP are endless. In this study, a webpage was designed providing drop down menus for the user to select the type of data to be presented and any possible calculations required. These could be from simple averages and percentages of simulation data, to more complex equations which the user could manually specify. The invoked results, could then be presented on a graph. For example, a simple graph could be produced by selecting to view the variation of incident solar radiation of the South facade on the X axis over the height of the building on the Y axis. The invoked tables from MySQL, could also be exported to other software as part of a report, or imported to EXCEL to reproduce the same graph or analyse that part of the data even further.

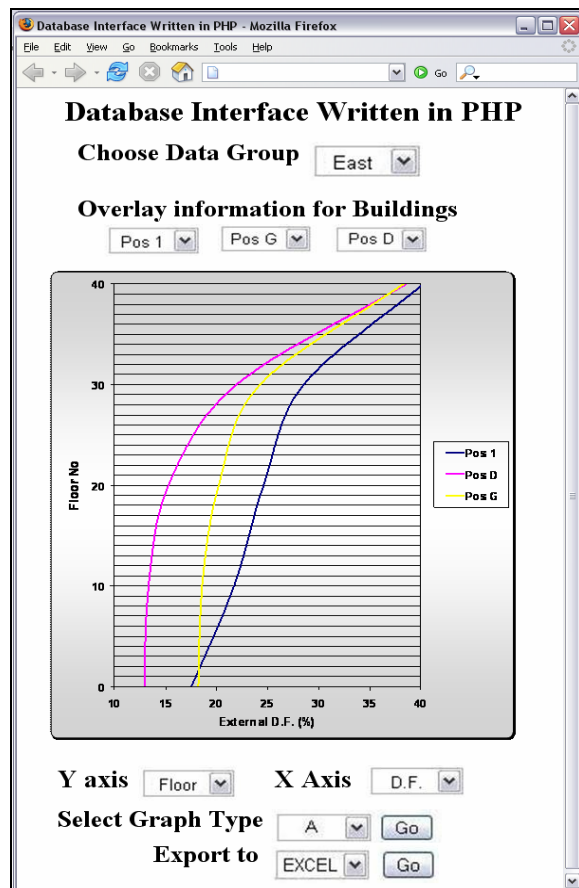


Figure 9 Example of a simple PHP page for previewing simulation results

The HTML webpage with PHP scripting code, can contain as much information as the user might need for analysing simulation data. The code written in PHP, could allow for any sort of analysis of data, may that be a simple average of values, or a statistical analysis.

Previewing data in such a way, allows the simulation team to get a better idea of the results produced thus far and spot any discrepancies in data, or trends which otherwise could be missed.

DISCUSSION

The generation, management and analysis of simulation data can have a number of benefits for a design-research team as well as their client. For the team undertaking the analysis, the use of scripting allows them to model and simulate the performance of parameters, that would otherwise not even be undertaken, due to the amount of time that would be required to complete the modelling. Furthermore, it allows the team to spend more time and concentrate on analysing the data, rather than just modelling the environment. For the case study undertaken, the generation of variations of the initial model (about 280 models), as well as the exporting and importing of simulation data, took less than 20 minutes to

complete. The RADIANCE runs and construction of the initial model in ECOTECT were considered as separate processes, since they depend on the users' ability to model an environment and on computer processor speed.

Generating such a huge number of data, allowed the simulation team to model in great detail all those parameters necessary to complete the study. As a result, simplifications were avoided and a more accurate representation of the actual performance of the building could be drawn.

The use of on-line databases through a webpage interface also has a number of benefits. Firstly, it provides a central location for storing large amounts of data. More data could always be added at any stage of the analysis process. Individual members of the team could check on the results of other people, thus avoiding modelling the same thing twice. Furthermore, by having a standard output of records that cover all of the results calculated (such as shading masks of windows), future analysis on the same project would save a lot of time, by referring back to this database.

For the case study undertaken this proved to be very useful, since the analysis was extended to include the adjacent buildings as well. Having a framework in place for generating and storing all that new analysis data, allowed the simulation team to concentrate on the analysis of the results, rather than trying to cope with the additional modelling.

The HTML with PHP webpage provides a simple and userfriendly interface, that people without any experience in databases, or scripting can interact with and graphically preview analysis data. This, significantly increases the number of people who can actively get involved into the project without having prior knowledge of the simulation methods. Figure 9, shows an example of one of the webpages created for this project, where the analysis data for three different buildings could be overlaid on the same graph and compared by any member of the team.

All this, allows the design team to check the analysis data of individual members of the team by previewing the results on the web, while getting a clearer idea of what the results look like, checking for any discrepancies and thus saving time in tracking any simulation errors, as well as tracking their progress on the project. The same applies for a supervisor or collaborator to the project, by looking at the results through the web from another location in the world.

Finally, these on-line tools are available for free allowing the user to experiment and explore their capabilities to best suit his needs.

FUTURE WORK

Simulation data generation through scripting and on-line data management and analysis can have numerous applications, apart from simply modelling the daylight availability of high-rise buildings. They could equally be used in thermal, acoustic and airflow analysis. The table format of each record would have to be altered to suit the needs of this analysis, but that could easily be changed.

Another area in which this whole system of generating and managing data can be used more is tracking the progress of a project. Standard scripts could be set up to test a series of parameters periodically throughout the design or simulation process. Thus, if a designer has adjusted part of the design, for example a shading device thinking that it is visually more pleasing that way, the next time the script is run and the results added to the database, it can become immediately obvious that the lighting and thermal performance has been adversely affected because it would show up as a dip in the on-line progress graph.

Future work is planned in the above areas as well as in the visualisation of data online. So far, the analysis data generated can be represented either as tables or as two dimensional graphs. The representation of data in three dimensions is the next step to the analysis of data. This way, anyone could preview the results online through a freely available three dimensional interface, thus being able to produce 3D graphs and also relate in 3D the location which the data correspond to, through a rough version of the model itself.

CONCLUSION

The method developed for generating and managing simulation data, requires a careful consideration into organising the data into a convenient and flexible format. The use of ECOTECH in this method, largely determines the way in which data are stored. At this stage, the other factor determining the way data is organised is the requirements of each individual project.

However, with the continuing development of this method and the integration of other software, it is likely that a set of standard formats will develop.

REFERENCES

- Ierusalimschy, R., Figueiredo, L.H., Celes, W. 1996. Lua, an extensible extension language. Software: Practice & Experience 26(6):635-652
- Marsh, A.J. 1996. Performance Modelling and Conceptual Design, International IBPSA Conference, The University of New South Wales, Sydney, Australia.

Ward, G.J. 1994. "The RADIANCE Lighting Simulation and Rendering System," Computer Graphics (Proceedings of '94 SIGGRAPH conference), July 1994, p. 459-72.

Wilkinson, B., Allen, M. 2004. Parallel Programming: Techniques and Application Using Networked Workstations and Parallel Computers, 2nd edition, Prentice-Hall Inc.