# IMPORT AS: INTERPRETATION AND PRECISION TOOLS

P. NICHOLAS AND M.BURRY
*Royal Melbourne Institute of Technology*
*Melbourne Australia*
*paul.nicholas@arup.com.au*

**Abstract.** This paper presents research on the relationship between digital tools and design communication, focussing on the interaction between architectural and lighting design. Early design integration often involves negotiating between different levels of resolution to inform a design that is still in formation, and part of the challenge is doing so in a manner appropriate to that phase of exploration. This paper describes some of the technical and social issues of translation and reports a project in which a generative design process supported the interaction between architectural design and lighting analysis; domains in which geometry is not necessarily a common ground.

## Introduction

Traditionally, designers have used the same methods to design as they have to communicate that design; plan, section and model being the most common. Increasingly, this is no longer the case – design techniques, understood as the tools we use and the methods by which we use them, have progressed rapidly while the methods for design communication, though now digital, have remained stagnant. At the same time, the purpose of design communication is shifting, from that of dissemination to that of integration. Increasingly, the desire is to integrate external knowledge into the design process so as to inform it at an early stage.

The translation of information between design and analytic domains can be made more or less difficult by the representations used to communicate that information. These representations form the interface by which both parties interact. In the case of architectural design and lighting analysis, the mapping process has been more difficult, requiring significant rework, 'cleaning' and interpretation on both sides. Practically, the result of this is that analytic tools are typically used late in the process to confirm or deny a particular solution, while intuition and precedent guide early design iteration. Drawing has not provided a particularly good interface – indeed, Peter Rice has argued that many environmental loads, including light, cannot be adequately addressed by drawing, and attempts to understand them through drawing are likely to be fundamentally flawed (Robbins, 1994).

This paper reports the use of a generative design process to intersect 'high resolution' lighting analysis with 'low resolution' design exploration. The process described determines the optimum placement and sizing of window openings in a façade for a program configuration that is still open to manipulation. Working from the premise that, to a large extent, integration is a function of the efficacy of the mapping technique employed [mapping being the term used within practice to describe the translation and interpretation of information between programs and between domains] it presents evidence that mapping can beneficially be used as a constructive act that extends beyond the replication of geometry in semantically different domains.

## Mapping: relating design and analysis

An early claim for the introduction of digital tools to the design process was that their increased precision would solve the problem of description (Mitchell, 1999); greater exactitude would eliminate miscommunication and result in an integrated design process. But this has not occurred, and the complexity of integrating design and analysis, specifically the role of interpretation within this process, has been the subject of ongoing research and literature. Several key issues have been identified:

A. That standard mappings do not support design simulation communication, which are typically highly project specific and require 'expert translation' (Augenbroe,2001),

B. Interpretation rather than precision has limited design integration (Luebkeman,1992) , and that

C. Successful integration may require transformation: design and analysis frameworks may both need to be active for successful integration, rather than the typical situation where one is active and the other reactive (Johnson, 2004).

The focus on interpretation at the interface is twofold. There is firstly a technical problem; that typically CAD and analytic programs are semantically far apart. This translation is not well supported by existing mappings and therefore geometry useful in one deployment is not immediately useful in the other. A second problem is that information returned from analysis also requires 'expert translation' and interpretation to be developed within the design intent. For instance, analysis might reveal that a specific part of the façade receives a lot of sunlight throughout the year: should this mean larger windows or a greater extent of shading, a change in façade form or cladding or a change in the building's programmatic layout?

## Mapping: export as

Both engineers and architects are to an extent limited by the capacity of their tools for interpretation. CAD and analytic software typically affords a number of ways of saving a file, both through the various native file-types and exchange formats, through an 'EXPORT AS' type command. Generally the user chooses whichever file-type is native to the program that they will be exporting to, if available, or else chooses a certain file-type because it provides some particular affordance.

CAD programs do not have the explicit inverse of this command, 'IMPORT AS' – all importing is done through the all eggs in one basket 'IMPORT' command, where whatever information within the file that is semantically compatible with the program is extracted in a single way, and that which is not, ignored. Contrast this to opening a text file [.txt] with *Excel*, an exercise in constructing a mapping that may provide some insight into how an 'IMPORT AS' command in CAD might function. Via the import wizard, the user can determine exactly how the information within the text file should be mapped to the *Excel* spreadsheet: on what line the translation should begin, whether the relationship of text strings to cells should be determined by fixed widths or delimitation, how those delimiters should function, whether the cells should be formatted to reflect dates or times etc, etc. There are a surprisingly large number of possible string to cell mappings.

## Mapping: in practice

The [self-described] workflow of an analyst using *Radiance* (*Radiance* is an industry standard ray-tracing tool used in lighting analysis) is shown in *Figure 1*.

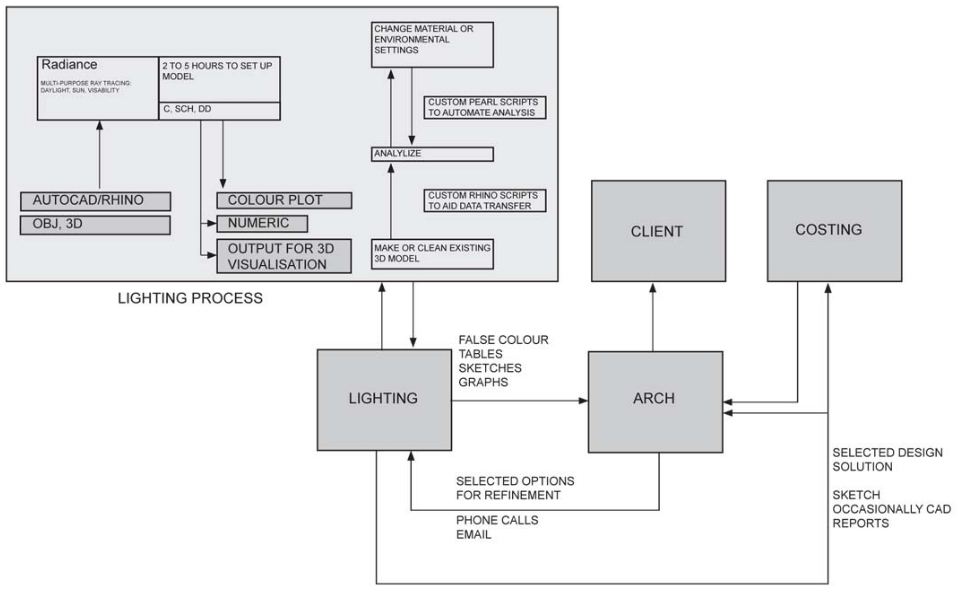

Radiance
MULTI-PURPOSE RAY TRACING: DAYLIGHT, SUN, VISABILITY
2 TO 5 HOURS TO SET UP MODEL
C, SCH, DD
AUTOCAD/RHINO
OBJ, 3D
COLOUR PLOT
NUMERIC
OUTPUT FOR 3D VISUALISATION
CHANGE MATERIAL OR ENVIRONMENTAL SETTINGS
CUSTOM PEARL SCRIPTS TO AUTOMATE ANALYSIS
ANALYLIZE
CUSTOM RHINO SCRIPTS TO AID DATA TRANSFER
MAKE OR CLEAN EXISTING 3D MODEL
LIGHTING PROCESS
CLIENT
COSTING
FALSE COLOUR
TABLES
SKETCHES
GRAPHS
LIGHTING
ARCH
SELECTED DESIGN SOLUTION
SKETCH
OCCASIONALLY CAD
REPORTS
SELECTED OPTIONS FOR REFINEMENT
PHONE CALLS
EMAIL

*Figure 1.* Modes of information transfer: Radiance within a typical design process

It can be seen that while playing a significant part in the process, digital tools and techniques play little role at the interface between architectural and engineering domains. The transfer of information takes place primarily through verbal or graphic means, including false colour images (a technique for visualising datasets), tables and graphs, and sketches. This information is typically packaged as 2D images and graph within a report. The work of relating the design and analytic frameworks is carried out entirely by the engineer, and only a limited amount of intelligence is passed between the parties, making quick, iterative design difficult. For many types of analysis this has partly been a product of history, a result of the tools being so hard or 'clunky' to use.

## The design problem

The design problem developed from a recently conducted interview, in which a description was given of the process undertaken to design a semi-porous skin covering an existing art gallery. The design intent was to create a semi outdoor space with its own microclimate, and one of the important design questions was "what were the optimum properties for the skin in terms of sizing, location and quantity of the holes?" In describing the role that digital tools played in the development and communication of the design information between architect and engineer, the interviewee, a mechanical engineer, stated that *"The tools are useful to produce images, but when it comes to things like how a space is actually affected by a design they're no good… The question the architects were asking was 'how big do the wholes need to be, and where should they be?' We couldn't answer that with colour plots."* This project was particularly complex, given that many environmental factors were considered and simulated. However it would seem likely that there was enough information produced to have enabled a more direct relationship between analytic results and geometric representation. But, as the following section details, this is not such a straightforward process.

## Low to high resolution

The following sections detail the process of designing an optimised façade system, where the optimum location and sizing of openings and diffuser panels are generated directly from analytic information. The design proposal, for an inhabited bridge, contained a complex programmatic mix of gallery spaces, cafeteria, administration areas and workrooms etc, each of which required differing levels of natural light. Being in the early stages of design, where the configuration of these programs was not yet determined, led to a particular problem in the relationship required between design and analytic frameworks: different program configurations needed to be tested quickly without the need for reanalysis.

The CAD program involved was *Rhino* [with a later *Generative Components* alternative], and the analytic program was *Radiance*, an industry standard simulation tool for ray-tracing. The design process is a generative one, in which generative design is understood as the use of a structure, often in the form of computer code that utilises rules, variables, and external information to generate geometry. The designer designs the code, of which the geometry is an outcome, and by altering the inputs and relationships within the code can generate a large range of possible designs. In this case, the range of possible designs represents the total number of possible program configurations within the building, and their corresponding optimal window opening configurations.

## Cad to Radiance information transfer

An initial unarticulated building envelope was designed within Rhino. This envelope was split into 750 X 750 mm polygons, each a possible site for a window opening. This grain was chosen because it was fine enough to stop the sunlight coming through any particular panel from flooding the interior, and it sat well within the range of standardised building components.

To accurately and efficiently reproduce polygons and the information associated with them in both CAD and analytic environments, it was necessary to retain a link between each façade polygon and the analytic results associated with it. With over 1300 possible positions for openings, it was essential to understand how to manage the large amount of information in an orderly way as it undertook two translations: from *Rhino* to *Radiance*, and from *Radiance* via *Excel* [where the analytic results were collated in a design file] to *Rhino* [or *Microstation* and *Generative Components*]. Tools that support this translation were found to be non existent, as analytic information of this kind is used almost exclusively to produce visualisations either within the software or via programs like *ParaView*.

A rhinoscript was written that, for any given collection of n-sided meshes within *Rhino*, output 5 text files, each coordinated so that the information pertaining to any particular mesh always occurred on the same line in each file. The benefits of writing an export script rather than using a pre-existing file-type were:

A. Elimination of cleaning. The term 'cleaning' refers to the task of manually rendering a 2D or 3D CAD file importable for analysis. It typically involves stripping out superfluous information, geometric rationalisation and simplification, re-layering etc. to create an idealised version of the information.

B. Pre-application of materiality. Layering within Rhino was used to assign simple material definitions to geometry. By using strings recognisable to Radiance this intelligence could be carried over into the analysis model.

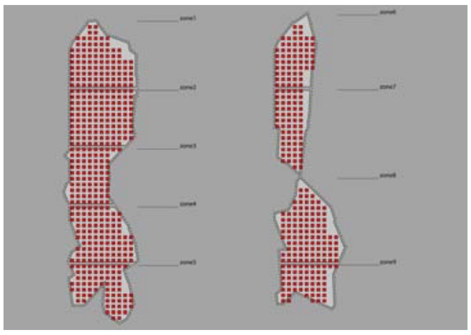
C. Control over naming. Naming each geometric entity provided a means to synchronise geometric entities with the analytic information associated with those entities.

D. Knowledge made explicit. Radiance is an idiosyncratic program, in that it is driven with a high level of customisation. Working closely with the analyst led to an understanding of exactly what information was required, and how to communicate that most effectively.

## Relating the Frameworks

This section discusses how the analysis and design frameworks were related through the method of simulation, the collation of the results and the way by which those results provided the input data for geometry generation. Work in matching these frameworks had already begun, as detailed above, and one effect of this was that the model for simulation was geometrically the same as the architectural model. While this avoided the issue of multiple geometrically dissimilar models, it is important to note that there is no actual advantage in having the same geometry and in many cases it would be beneficial not to.

Because the design intention was to explore a solution space of possible program configurations, rather than analyse a single specific case, the analytic framework was structured so as to analyse the building as a series of generically programmed zones. By establishing the relationship between each zone and the façade, programming could later be made specific by the designer who could explore different programmatic possibilities by changing which program occupied which zone. The designer could then generate the optimum façade openings for that combination without the need for re-analysing.



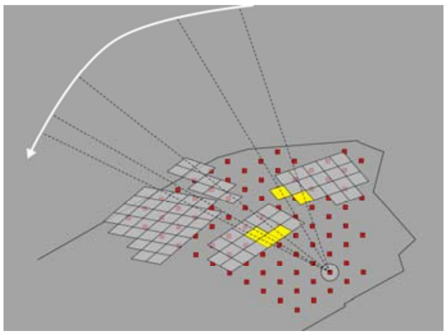
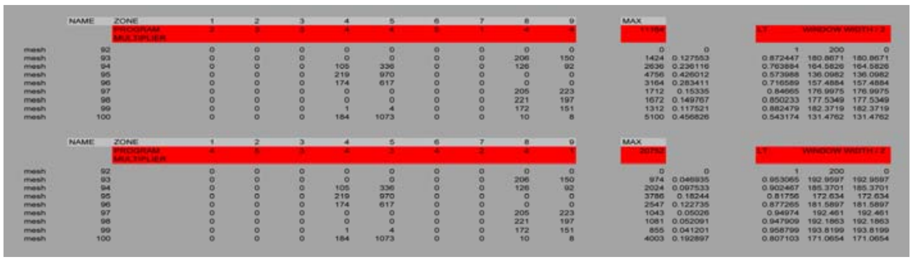


*Figure 2.* Zoning and sampling points *Figure 3.* Simulation for a single point

The floor-space was divided into nine programmatic zones, each approximately 100 m2. These zones were filled with a 1X1m grid of points, which became the points tested in the *Radiance* simulation (*fig 2*). A 1X1m grid of points was seen as giving a sufficiently accurate result: the more point locators the finer grained the results are, and the longer analysis takes.

Within the *Radiance* simulation, each point casts rays towards the 'sun' [a collection of its combined positions over the course of a year]. This amounts to thousands of rays for any point. When a ray passes through a façade mesh, a hit is registered for that mesh (*fig 3*). The more hits registered by a mesh the more direct sunlight reaches the point being analyzed via that façade element. Through this method the relationship, in terms of direct sunlight, between any particular façade element and any particular location within the building can be determined. Any single facade element typically affects multiple zones.



| NAME | ZONE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | MAX | | LT | WINDOW WIDTH / 2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PROGRAM MULTIPLIER | 2 | 3 | 3 | 4 | 4 | 5 | 1 | 4 | 4 | 11184 | | | | |
| mesh | 92 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 200 | 0 |
| mesh | 93 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 206 | 150 | 1424 | 0.127553 | 0.872447 | 180.8671 | 180.8671 |
| mesh | 94 | 0 | 0 | 0 | 105 | 336 | 0 | 0 | 126 | 92 | 2636 | 0.236116 | 0.763884 | 164.5826 | 164.5826 |
| mesh | 95 | 0 | 0 | 0 | 219 | 970 | 0 | 0 | 0 | 0 | 4756 | 0.426012 | 0.573988 | 136.0982 | 136.0982 |
| mesh | 96 | 0 | 0 | 0 | 174 | 617 | 0 | 0 | 0 | 0 | 3164 | 0.283411 | 0.716589 | 157.4884 | 157.4884 |
| mesh | 97 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 205 | 223 | 1712 | 0.15335 | 0.84665 | 176.9975 | 176.9975 |
| mesh | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 221 | 197 | 1672 | 0.149767 | 0.850233 | 177.5349 | 177.5349 |
| mesh | 99 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 172 | 151 | 1312 | 0.117521 | 0.882479 | 182.3719 | 182.3719 |
| mesh | 100 | 0 | 0 | 0 | 184 | 1073 | 0 | 0 | 10 | 8 | 5100 | 0.456826 | 0.543174 | 131.4762 | 131.4762 |

| NAME | ZONE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | MAX | | LT | WINDOW WIDTH / 2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PROGRAM MULTIPLIER | 4 | 5 | 3 | 4 | 3 | 4 | 2 | 4 | 1 | 20752 | | | | |
| mesh | 92 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 200 | 0 |
| mesh | 93 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 206 | 150 | 974 | 0.046935 | 0.953065 | 192.9597 | 192.9597 |
| mesh | 94 | 0 | 0 | 0 | 105 | 336 | 0 | 0 | 126 | 92 | 2024 | 0.097533 | 0.902467 | 185.3701 | 185.3701 |
| mesh | 95 | 0 | 0 | 0 | 219 | 970 | 0 | 0 | 0 | 0 | 3786 | 0.18244 | 0.81756 | 172.634 | 172.634 |
| mesh | 96 | 0 | 0 | 0 | 174 | 617 | 0 | 0 | 0 | 0 | 2547 | 0.122735 | 0.877265 | 181.5897 | 181.5897 |
| mesh | 97 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 205 | 223 | 1043 | 0.05026 | 0.94974 | 192.461 | 192.461 |
| mesh | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 221 | 197 | 1081 | 0.052091 | 0.947909 | 192.1863 | 192.1863 |
| mesh | 99 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 172 | 151 | 855 | 0.041201 | 0.958799 | 193.8199 | 193.8199 |
| mesh | 100 | 0 | 0 | 0 | 184 | 1073 | 0 | 0 | 10 | 8 | 4003 | 0.192897 | 0.807103 | 171.0654 | 171.0654 |

*Figure 4.* Design file – 2 options

The results of this analysis were collated in a design file within *Excel* (*fig 4*). The design file is a database cataloguing each façade mesh and the number of hits recorded by it. Within the design file, the designer can test different programmatic configurations, in this case 6 programs types between the 9 zones. This combinatorial testing is achieved by the application of a multiplier specific to each program, which relates the amount of daylight reaching that program to its required daylight factor as specified by standards.

These calculations are used to determine the sizing of façade openings and diffusers. The results are filtered through 8 possible window sizes, which range from 100 to 800mm in width and step in 50mm increments. If the level of sunlight is calculated as being too sensitive for a particular program, or of no benefit, no opening occurs. Behind the spreadsheet sits a *Visual Basic* routine that takes the calculated window width as an input for the geometry generation in *Rhino*. The script generates on average 1000 correctly sized and located window openings for any tested combination. In the *Generative Components* version, it also generates the diffusers, taking advantage of GC's capability for mass instantiation of parametrically defined objects.

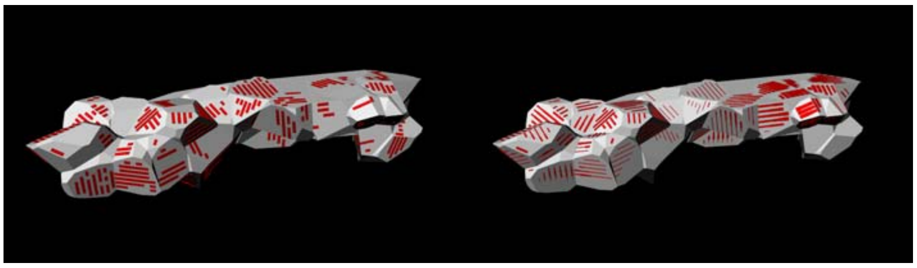


*Figure 5. Two configurations from the design file.*

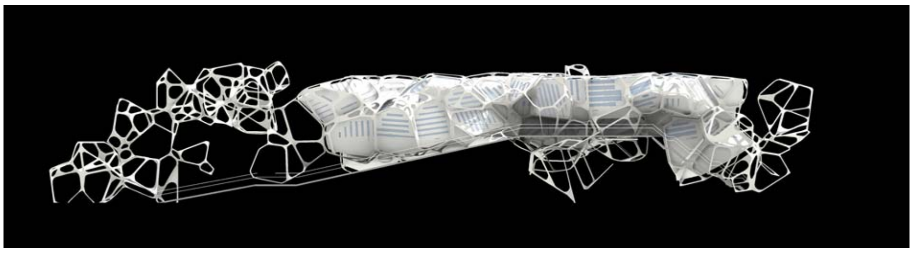


*Figure 6.* Façade openings instantiated on architectural model.

## Conclusion

Seeking to negotiate different levels of resolution is characteristic of the design process, but difficult to achieve. The project reported in this paper has described how digital strategies of practice can be used to construct such a mapping. It has presented some of the reasons why this extends beyond the direct translation of geometry in semantically different domains, the most significant being that analysis often involves modelling the problem rather than the building. This is particularly the case in seeking to integrate architectural and lighting design, where the issues involved are often not geometric.

The generative process described had the benefit of both quickly informing a design in a manner appropriate to early design exploration and of extending the inter-domain understanding of the parties involved. In comparison to more common 'over the top' modes of communication, such a process can avoid later re-engineering by the earlier integration of analysis, which assumes an active rather than reactive position within the design discussion

## Acknowledgements

The author gratefully acknowledges the contributions of Tai Hollingsbee and Jon Morgan of Arup Melbourne

## References

Augenbroe, G, Building simulation trends going into the new millennium, in Seventh International IBPSA Conference, Brazil, 2001

Evans, R.: 1997, Translations from drawing to building, The MIT Press, Cambridge.

Johnson, S., von Buelow, P., Tripeny, P.: 2004, Linking Analysis and Architectural Data: why it's harder than we thought, in *Proceedings of the 23rd Annual Conference of the Association for Computer Aided Design in Architecture*, 230–243.

Luebkeman, C.:1992, Good Fences Make Good Neighbors?, Architronic, 1(1.07)

Mitchell, W.: 1999, A Tale of Two Cities: Architecture and the Digital Revolution, *Science* 285(5429), 839 – 841.

Rice, P.:1994, An Engineer Imagines, Ellipsis London Ltd, Hong Kong.

Robbins, E.: 1994, Why Architects Draw, The MIT Press, Cambridge, MA.