Abstract. In education and research at this time there is arguably no longer a need to build a strong case for the power of CAAD to support designers – the evidence is there. The major challenges no longer centre on hardware, software and graphics potential or on skills acquisition and adoption. The research that we will report here reveals that computational complexity and geometrical complexity are emerging as the sharp issues that demand a major review of how we model the large hybrid spaces that we seek to construct in design. Computational relational modeling and scripting may have opened a trove of creative possibilities. But it may delude us into painting ourselves into a corner: infinite variety within a much reduced palette of opportunities.

Keywords: Architectural Geometry; relational modeling; parametric design.

Introduction

There is value in focusing on the ways in which the flexibility of our more sophisticated modeling tools can also be paradoxically constraining. In this paper we will support this position through examples. These are taken from practice-based research and real application as well as student projects operating with fewer real-world constraints in order to reach some conclusions about the more successful ways of engaging our spatial intuition and identifying valuable directions for future research.

Example 1: the human optimiser

This example illustrates the way that we use software tools to extend the dimensionality of the space in which we design in real time while avoiding the impossible mathematics of multidimensional problems by engaging human sensorial abilities to do the solving.

This is a relatively simple model of a shade/shelter structure undertaken in a final year undergraduate architectural research course at Queensland University of Technology (QUT). It is conceived as a series of pivoting curved segmental shells. The shells
can revolve to modify the form and hence the natural lighting and weather exposure within the shelter through the seasons. Whether or not the construction issues would be simple to resolve for this kinetic structure (we can argue that the technology of children’s playground equipment provides some leads), the initial model is clear in it’s intent, has relatively few variables and all of these are directly related to the geometry or shape information about the individual shells and their relationship to one another.

This model was built using Bentley Generative Components™ software. There were eleven related variables that the author, Young Ji Seo (Ellie), regarded as important to controlling the shape, orientation and relative positioning of the shells. The shells are arrayed around an arc of variable radius and angle, the number of shells into which this arrangement is divided can vary. The overall disposition of the shelter can vary in these three ways (radius, overall angle of arc, and number of shells.) Meanwhile each shell has 8 further degrees of freedom. The author has decided that the arrayed shells should be similar, a single repeated component, so each of the internal variables determining the shell shape is linked to a value common to all. The shell is a feature (a custom built geometrical assembly that has been added to the available library of types).

The shell has an angle that controls its curvature in the vertical direction, in other words, the shells can collectively rise from the base, splaying outwards like the skirt of a spinning ballroom dancer.

Staying with the skirt metaphor, they can be drawn up from the ground to reveal more. They can be drawn down from the centre to create a wider central oculus. There are three variables that control curvature, one drawing the form down and more hunched, or taller and straighter, another allowing it to belly outwards or slim down, a third that leaves the central height where the shells meet unaltered but progressively hunches their ‘shoulders’ upwards. The shells have a variable angle of inclination across their width to allow each to lap with its neighbour without clashing and finally, each shell can rotate on a pivot axis, opening up the shelter to the sun and weather.

Following this long and metaphorical description of the geometry of the shelter, it is clear that there are eleven chosen ways in which the model can be changed. Ten of these control shape in ways that, depending on detailed construction, are probably relevant only during the design process, the eleventh allows the simulation of the operation of the dynamically opening structure. Within this simple and geometrically constrained schema, there are eleven degrees of freedom. It is what Roger Penrose would refer to as an eleven dimensional space (Penrose, 2005).

It is possible by choosing combinations of variable values within certain ranges to create a form in which the shells clash, losing the ability to rotate freely. Even without introducing the complexities of external parameters such as sun angles to drive the geometrical variables through defined relationships, how could you ever discover mathematically where these regions in the design space lie with eleven variables?
This model is a tool for the human solver. Each of eleven variables, whether created in the shelter model or in the shell feature is defined as a graph variable, in other words, its value within a certain range, with defined increments is linked to a slider bar and all eleven slider bars as displayed within a single ‘control panel’. The user can literally drive the model in real time to explore the design space, smoothly varying any combination of the eleven variables.

Having set up the system using numerical ranges for the values, by way of constraints, it is possible to enter a dance with the form that is haptic and metaphorical without ever creating a shelter that is too low to inhabit or too curved to fabricate, and in which clashes can immediately be detected by eye.

Example 2: Sala Creuer

In relational digital models all the geometrical elements are associated to one another or to higher order schematic geometry linking the parts – the ‘schema’. In this way, when a change is made in one particular area, the whole model has the potential to respond by updating in response to the new criteria or values while maintaining the relationships, as long as this is geometrically possible.

It is a common catch cry of those working in architectural practice who have experience of constructing relational digital models that this is a good workflow for modeling a project once past the very volatile conceptual stage but where the design is still undergoing significant refinement and iteration. It is not only too slow and cumbersome for the very early stages of design, especially at the stages of ideation or conceptualization, but the level of change in conceptual design is so fundamental that it can never conform to a relational schema. No relational map will be sufficiently flexible to model, say, ten completely different proposals for a site in the earliest stage of design exploration within existing parametric design software paradigms.

The debate quickly descends into one of semantics with no useful outcome. In theory, a well
organised designer will structure their schema from a ‘verified concept’ so completely that the iterative refinements ripple through the relational database with ease and facility. For engineers perhaps, for whom parametric design software was originally intended, such order and method is a reasonable prospect for a ‘tame’ problem solver. But the messiness of design or its inherent wickedness as a problem space by definition thwarts such otherwise intentional clarity (Rittel et al 1973). No sooner has the designer completed the schema when fresh information or fresh inspirations (or both) intervene to corrupt the enterprise. Erasure and redrafting, the enemy of efficiency and disrupter of the flexible (parametric) model re-intervene as an essential ‘modus operandi’. The research question becomes: ‘can we sensibly model architecture using parametric modeling if inevitably the process involves erasure and redrafting?’

Our tentative conclusion is to employ the usual obfuscation: ‘horses for courses’. For complex geometries hanging off complex schemas, it is by no means clear that sketching (because that is what we are referring to) is efficiently conducted using parametric modeling software – such software places extra demands on the user in obliging them to declare information up-front. A ‘parametric point’ has more information than an ‘explicit point’ in space’. In the early years when parametric design software was first experimented with for architectural design, it appeared to be inevitable that the designer initially tried to destructively construct the model as an essential first iteration in the design motivated acutely towards establishing the schema; a kind of trial by failure. Only once the limits of the desired flexibility were consolidated could the white gloves be worn for the actual robust model that could be persuaded into its desired outcome. While successful, this process is predicated on sole authorship: the designer first constructs the schema, and then constructs the model (Burry, 1996).

Our current findings through action research point to more complex relationships between the need for speed and efficiency, and the need to reflect. Where it is possible for the sole author to perhaps construct and reconstruct a sketch design using parametric software if that is their preference, it is most unlikely they can share the model with a colleague, because for their schema to be readily intelligible by others, it would have had to be structured very carefully, an orderly approach that is inconsistent with any workflow reliant on emergence within the design process.

An example of using parametric modelling only for late refinement after repeatedly modeling in explicit 3D modelling software is the [Sala Creuer space] above the central crossing of the Sagrada Familia Church nave and transepts. This is a live project for which documentation commenced on April 1, 2008, and work on site will commence in April 2009. Design refinement continues. The five chronologically arranged images showing the design development of just one element of the project demonstrate the futility of trying to develop the design around a schema, and the example is a single window from a much more complex arrangement.

The Sala Creuer is the space above the vaults over Gaudi’s Sagrada Familia Church crossing. It is visited by the public and is the node of convergence

Figure 3
Chronological sequence of five variants of the design of a window element in the Sala Creuer space in the Sagrada Familia church
of the circulation from the lower parts of the building to the lift and stairs that serve the central tower. The space is like a drum with a ‘floating’ auditorium within, surrounded by a ring of windows. At each cardinal point in the plan there is a principal window based on an original by Gaudí. The central windows placed on each axis are flanked by a lateral window which is a version of Gaudí’s original reduced in size and geometrically driven detail. Over the last twelve months there have been numerous iterations. All conform with the genotype but each has too many significant departures to be a geometrically derived version of any other. There is a kit of geometrical components with which each version conforms, but unless each possible component is scripted as an overlay to the parametric model, there is no real value in a relational model.

The point is that as the architectural design community becomes more familiar with parametric design software, there needs to be an awareness of the implications in its use. The schema as a design construct paradoxically needs to loosen-up. Some of the packages available offer such opportunities through having a scripting interface. If not actually shifting the goalposts at least we can widen them; we can mitigate the domination of the highly structured schema that orders the design to a more benign and flexible scripted narrative.

**Example 3: Pointland**

This example is a design that seems extremely simple conceptually, easy to visualise and to conceive of possible variations. It is also comparatively easy to model physically. The difficulty the student author encountered was with parameter values that needed to be able to lie in the range from zero to something where a zero value implied that some part of the geometry was effectively non existent, while supporting other geometrical dependencies; for instance an arc of no radius, a line of no length. We can conceive of such a thing but it challenged the logic of the particular software.

The project was undertaken as part of a multidisciplinary flexible modelling elective. Unlike the first example this proposal was abandoned in favour of a very different project due to difficulties that illustrate the importance of semantics inherent in the hidden algorithms in software.

This student chose to emulate in his model the Mobils pods© by David Douglas Haring to investigate how other modules might be developed within the same language. These are simple lightweight framing rooms, open to the view at each end and formed as a well rounded filleted rectangular tube on short legs in which two or three people might gather or the view might simply be framed.

The idea of the student project was, staying within the original language of the creator (a tricky syntactical question), to elaborate the simple space, molding the walls to provide additional functionality – a table surface or seat for instance. This raised many issues, not least, at which point the language of the Mobilis pod© was lost. Perhaps this occurred with the first move to elaborate the simple form.

The aim was to create a single digital model that could beget a series of customised pods for different settings and applications by simply changing
numerical parameter values, as opposed to revisiting the construction of the model. The external dimensions and notional profile would be maintained, while the internal surface could be deformed to provide seating or tables. The variations were to include some in which fillet arcs or cylindrical sections between surfaces disappeared and became sharp angles or continuous planar surfaces in the process of changing from one variation to the next. Within the parametric software in use this transition from an arc of positive radius to an arc of zero radius proved impossible by any means attempted. A familiar workaround is to reduce the radius to such a small dimension that it is no longer perceptible but this is hardly a rigorous approach to modeling.

Is this a fundamental logical problem? The idea of going from the single dimension of a line or an arc to the dimensionless state of a point as the line of no length or the arc of no radius is potentially a troubling one. But no, it seems that it is possible within a purely logical construction to do this. Working in a different software, a graph variable may be assigned to a radius with a range that not only descends to zero but will comfortably interpret a negative value as its positive counterpart. Such semantic decisions within software are strategic and responsive to the primary context and use. They reflect the particular nature of objects within the particular geometric world of the software. Each software represents what is referred to in philosophy as a ‘possible world’.

This same problem taxed students charged with creating a model in which a cube or tetrahedron could change smoothly to a sphere, simply by adjusting numerical variables (as opposed to structural changes to the model.) This was adopted as a warm-up exercise from an earlier more involved student project (Burry et al, 2001). The first inclination was to approach this through inclusion of an algebraic expression controlling the relationship of certain points that travelled between the edges of the cube and the great arcs of the sphere. However, the quickest thinking or most mathematical notation - averse student quickly realised that by taking equivalent points on the two geometrical extreme cases (the polyhedron and the circle), joining these equivalent points by lines and using points on these lines, their position on the line controlled by a ratio parameter of the lines length to define the ‘edges’ of the form, they had a form that transformed by varying the value of a single ratio parameter. When this value was zero, there was a cube, when one, a sphere, between these values a half way form. By going below zero the cube edges were sucked in to make spiky form of convex curvature. By going above one, the sphere began to develop extrusions. In this way, they bypassed the soft semantic problems with the zero or negative dimensions of circles and lines.

**Discussion and Conclusions**

Design is all about simultaneous variation, many factors, qualities, decisions which are all mutually dependent in various ways. Design sketches are commonly crowded with variations and alternatives. Trial decisions are taken in order to explore a particular scenario and later overturned. Computation, in particular associative geometry or relational modeling, provides a structure within which to hold, simultaneously, innumerable variations and combinations.

We have used three examples, the first and third from undergraduate student research, the second from current work in practice that demonstrate three aspects of the application of applying digitally-supported relational modeling at different stages of design. They are simple examples in the sense that they represent comparatively small databases of relations and deal with comparatively prescribed design tasks – determining the surface shape and arrangement of a set of similar arrayed shells, determining the surface geometry of a stone window assembly and creating variable two dimensional profiles to control the molded shape of a pod. The geometry in each example if not homogeneous, belongs to a language; the relations are of similar logical type. Nevertheless they are rule-bound spaces that raise particular geometrical issues and create particular constraints.
The first example, the shade/shelter structure, highlights the continuing importance of the design-tool interface and mode of interaction. The software environment here supports easy interaction with the script and accessible, economical notation. However, tidy though the structure and creation of the space of design is, its exploration is not well supported by logic. The number of variables even in this very small prescribed example would allow very limited mathematical analysis. The human hand eye is still a very efficient way to explore and evaluate. This might not be the case in more hard-edged analysis, for instance finding those design outcomes that would achieve a given range of average lux level on the ground under the shelter at a given time of year and location. But in that case it is complex to optimize for multiple criteria (Roman et al, 1995). The light levels could not be linked to all the shape parameters simultaneously.

The second example, the Sala Creuer, investigates the real world reasons why relational modeling may fail to support the iterative process or is too time intensive in early design.

The third example, the Pod variants, is a small specific case that confronts the ontological specificity of software. Each software is in itself a design space, albeit generally a very large one, that comes complete with an inbuilt language and set of geometrical constraints and with it subtle but effective barriers to particular ways of working.

Hyperspace feeds an expectation that we can link anything to anything just as we may do in thought, literature, art or on the web. It is a rude awakening when our mutable, morphing digital model falls foul of the constraints of its representation in three dimensions, even three programmed virtual dimensions, where space is constrained by measurement. It has the advantage over kinetic physical prototypes such as those of the mechanical engineer through its infinite ability to change shape. However, the complexity of a large network of relations between the geometry of the parts makes it hard to predict the performance of the model, or even which parameter value combinations represent viable three dimensional manifestations. We are back in the empirical realm of the design sketch albeit at a higher level of representation of our intentions. Our examples are very small comparatively simple esquisses, how much more constrained is the relational building information model (BIM) if we attempt to model a building in its entirety as a map of relationships. Why are these relational design models seemingly so much more brittle and constrained than other networks with which we are familiar?

Parametric or flexible three dimensional digital modelling is sometimes referred to in the general category of constraint modelling. Decision making in design is tantamount to imposing constraints, limiting the size of the domain of possibilities. This mode of using the computer to record relations ahead of dimensions aims to reduce the severity of impact for each constraint, to defer precise form and space making through prioritising the formulation of a schematic graph of relations. However this graph of relations also takes a form or its own and the mutual impact of the constraints can become a complex system in which the number of variables defies mathematical modelling. It is quite usual to have to explore the space empirically by varying individual parameters. The complexity is exponentially increased with the database size and number of contributing authors. The relations within the graph are not homogeneous. They are in fact extremely heterogeneous. Each represents a specific and hierarchical (directed) link. If we create a set of points on a surface, we cannot then control the shape of the surface by means of those points as this would represent a logical cycle.

The danger that lies here is that the very flexibility and the complexity that it implies can become constraining. While the formal novelty and level of design resolution of some of the more innovative contemporary architecture projects are attributable to a high level of computational geometric sophistication, deferment and relational modelling are also valuable tools for the most pragmatic and
instrumental reasons. To be able to adjust column spacing for code compliance or change façade depth in accordance to new information in developed design seems, at first, a welcome opportunity to marry creative freedom to the need to observe external strictures. Considered more closely there is also the possibility that when this alleged flexibility is applied to the resolution of the whole of a large project in the building information model tradition (BIM), the apparent but illusory opportunity to model now and design later belies a powerful and normative force for darkness with regard to creativity.

**Acknowledgements**

We acknowledge Young Ji Seo, final year QUT architecture student 2008 and Salvador Pires, RMIT student 2007 for the opportunity to show their work and their contributed images and Prof Robin Drogemuller as joint supervisor with Jane Burry of the final year research students at QUT. The research reported here has been part funded by the Australian Research Council. We acknowledge their support and that of the Sagrada Família Church Foundation in Barcelona for the opportunities they provide for extending the work of Gaudí into contemporary architectural practice and research.

**References**


