



**Parametric and generative methods with building information modelling: Connecting BIM with explorative design modelling**

**Author**

Fernando, Ruwan, Drogemuller, Robin, Burden, Alan

**Published**

2012

**Conference Title**

PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON COMPUTER-AIDED ARCHITECTURAL DESIGN RESEARCH IN ASIA (CAADRIA 2012)

**Version**

Version of Record (VoR)

**Downloaded from**

<http://hdl.handle.net/10072/123866>

**Link to published version**

<https://www.caadria.org/conf/caadria2012/>

**Griffith Research Online**

<https://research-repository.griffith.edu.au>

## PARAMETRIC AND GENERATIVE METHODS WITH BUILDING INFORMATION MODELLING

*Connecting BIM with explorative design modelling*

RUWAN FERNANDO, ROBIN DROGEMULLER  
AND ALAN BURDEN

*Queensland University of Technology, Brisbane, Australia*  
*r1.fernando@qut.edu.au*

**Abstract.** Parametric and generative modelling methods are ways in which computer models are made more flexible, and of formalising domain-specific knowledge. At present, no open standard exists for the interchange of parametric and generative information. The Industry Foundation Classes (IFC) which are an open standard for interoperability in building information models is presented as the base for an open standard in parametric modelling. The advantage of allowing parametric and generative representations are that the early design process can allow for more iteration and changes can be implemented quicker than with traditional models. This paper begins with a formal definition of what constitutes to be parametric and generative modelling methods and then proceeds to describe an open standard in which the interchange of components could be implemented. As an illustrative example of generative design, Frazer's 'Reptiles' project from 1968 is reinterpreted.

**Keywords.** Building information model; parametric modelling; generative modelling.

### 1. The need for interoperability

With the rise in complexity that has come with contemporary architecture – whether it is to do with the complexity of form, the need for better performance, or facilities management – there is a greater need for tools to help the process. As the number of digital tools we use increases – there is a greater need for interoperability between different applications. Building Information Models are a way of allowing communication between different disciplines by having a semantically rich model to-and-from which data is transferred.

The open standard for building information models is the Industry Foundation Classes (IFC) which is regulated by the IAI (International Alliance for Interoperability). The specification is vendor-neutral and is in the process of becoming International Standard ISO 16739. In this research paper, all Building Information (BIM) models refer to IFC data models– which is always the case in practice as there are many proprietary applications that use their own internal format. One shortcoming of the IFC data model is that it does not allow for the exchange of parametric modelling data which this paper argues, can be very useful for many reasons.

## **2. The advantages of BIM and data-rich models**

Some of the advantages of BIM models shall be highlighted – information rich objects allow multiple disciplines to extract the information they need for analysis. For example, a curtain wall can be used by a light-simulation program – if the material properties were contained within the data. The model could be the same as the one created by the designer and analysed by the structural engineer.

Aish (2000) criticises the current fragmentation of design documentation as the result of the paradigm given by the personal computer where each file is a ‘discrete’ document stored as a file.

An advantage of interoperability – as opposed to trying to create tools on a single platform, is that the quality of the tools can evolve in their own disciplines. Parametric modelling, in many areas has focussed on geometry – and as a result of the objects being ‘immaterial’ it requires refinement before it can be used by other disciplines.

This ‘metadata’ can be an important factor in many practical cases. In design it is rarely the case that some parameter of a model is optimised without some opposing factor being simultaneously considered. For example when increasing the area of glazing or the thermal transmissivity of such an element – there is a cost involved in upgrading the glazing as well as the play-off between heat loss and visual quality – not to say anything about the aesthetic qualities of the façade.

There is a price to pay when it comes to using BIM – and that is the loss of freedom that is associated with geometry-only models, the uncertainty with how to ‘classify’ elements and the uncertainty by which elements are coupled to properties. It is hard to make general statements that hold true for all cases – but it should be considered that in practice, there are some cases (or patterns) that are more common.

### 3. What is parametric modelling?

Parametric modelling refers to many different concepts within modelling. The most general characteristic of parametric modelling would be that it produces more than static geometry – a model with a collection of primitive shapes (meshes, curves, points) unrelated to each other. Parametric models contain some form of structure that links elements together or recreates the steps used to create a model.

The most important aspect – for the future development of computational tools – is the embedding domain-specific knowledge into a formal system. This was recognised by Lee et al. (2006) and a lesson should be learnt from the field of computer science where current systems sit above multiple layers of older systems – and how higher level languages are used for greater expression.

There are five concepts associated with parametric modelling that this research is interested in:

**A. Explicit History** – also known as ‘procedural shape generation’ or ‘History-based models’ (Pratt 1998) in other writings. This is the most widely –spread use of the term parametric and software such as McKneel’s Grasshopper or Bentley’s ‘Generative Components’ allow for such modelling. These transactions usually take the form of an acyclic graph.

**B. Association** – which is also known as constraint satisfaction and is covered in the ISO standard for product design. To make advances in parametric modelling, it is essential that some control can be exerted over the sketches and profiles that are used by the model.

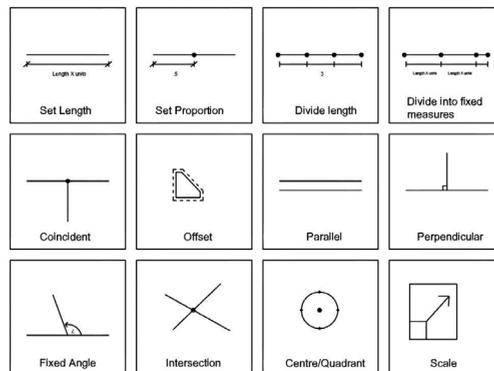


Figure 1. Constraints and modelling features.

**C. Types** – in programming, the word ‘inheritance’ is used as a formal mechanism for classes to derive their attributes and operators from a base class. This gives a ‘is-a’ relationship between objects. For example under the abstract class of ‘Person’ – two objects that inherit all the properties of

person could be ‘man’ or ‘woman’ – each ‘is-a’ person and can be used in a context where a ‘person’ entity is needed. In the case of modelling software an example is where a beam type can be replaced by any other *particular* beam because they both inherit from the same base class.

**D. Patterns and Perspectives** – In many industrial design tools, collections of parts are called assemblages. These are allowed their own variables which can modify some qualities of the assemblage. The manipulation of an assemblage might require a special ‘view’ or ‘perspective’ – in detailing the frame of a wall, an elevation is needed – a program manipulating spatial adjacencies might have an interactive graph control. The choice of terms ‘perspective’ was adapted from the work of Haymaker et al. (2004). It is used to describe a certain ‘point of view’ during the design process and can range from an orthographic drawing, a perspectival projection or a symbolic list. While it may seem a trivial point to give a name to something so basic to design, it should be noted that certain constraints related directly to the perspective where it is created. An example of this would be facade composition.

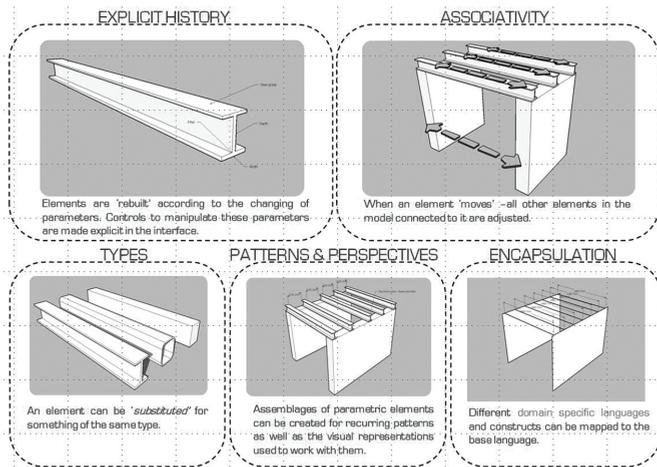


Figure 2. The five aspects of parametric modelling that this research works with.

**E. Encapsulation** – The features created for BIM work at a very low level – in order to become more expressive, there is the need to deal with more complex configuration using high languages. As an example consider the operations that could occur to a curtain wall – various parameters could be adjusted that change its properties. In a parametric model these operations are contained within the data-object. Encapsulation is essential for ‘intelligent’ objects.

### 4. Data in a building information model

The diagram below shows the various sets of information that is present in a BIM component. The two sets shown capture a generic list of properties and one definition that is particular to materials

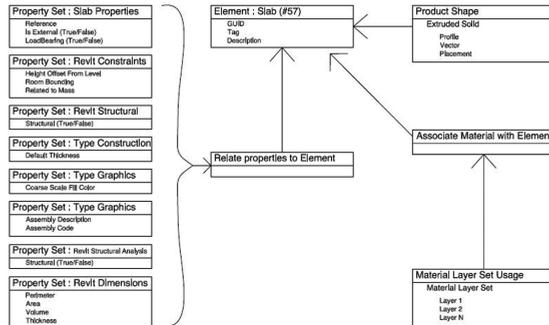


Figure 3. Property data within a model (based on IFC exported from Autodesk’s Revit).

Note how the object itself does not contain data – other data objects ‘point’ to it instead. With a parametric model, the geometry needs to be ‘rebuilt.’

As a result, there is a need for re-evaluation of constraints and a ‘geometry building’ process is needed. Figure 3 shows the three main components of the data carried in a slab – The generic properties, the material properties and the geometry. In this particular case, some of the default properties exported by Autodesk’s Revit are shown. If the model were adjusted to capture parametric data, then one possible implementation would be as follows:

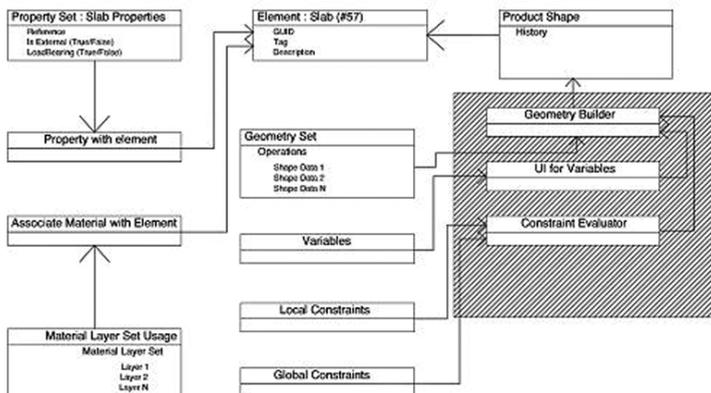


Figure 4. Elements of a parametric object.

## 5. Generative design methods

‘Generative design’ is a much looser term than ‘parametric’ modelling and can encompass any design process where a given ‘process’ generates some outcome that is then interpreted for its design merit. For illustrative purposes, a famous example ‘Reptiles’ designed and implemented by John Frazer in 1968 shall be used. This example is illustrative because it set the foundation for later work including the use of evolutionary tools for optimisation – field that is very topical today.

The ‘Reptiles’ (or REPetitive TILing’) described a system of folded plates made from tetrahedral units that encompass enclose a structure. This system had various advantages over the enclosure systems that were known at the time. As the Reptiles article from 1974 described – most of these systems have trouble when dealing with existing orthogonal conditions. The system consists of two types of folded plates one with the vertices of two faces of a tetrahedron and the other similar but with an angle of 90 degrees between the edges. Each of the tiles was described using coordinates in a 3 axis grid:

*The unit is located in a 60 degree grid with simple whole number ‘A’ and ‘B’ coordinates (range -9,99) ... A ‘D’ co-ordinate describes depth of the unit. The type and orientation of a unit are described by two digits, the first giving the type and vertical orientation, the second the orientation in the horizontal plane.*

*This whole description of the unit in space takes the form, for example, -66, 32 24 21, and is packed into one word of core space.*

Reptiles – Frazer (1974)

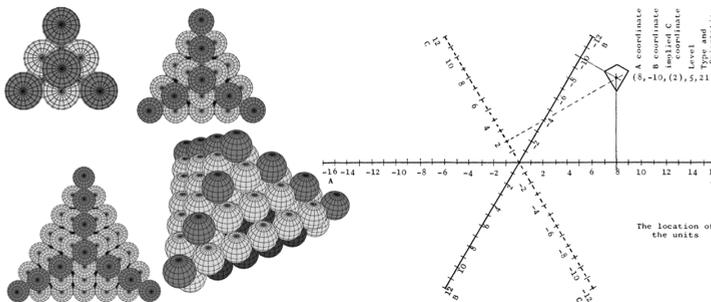


Figure 5. Iterations in the growth process and image from Frazer's Reptiles (1974).

One of the interesting features of the program was that it began growth from a seed – two particular seeds the ‘knot’ and the ‘star’ were given and from them many variations formed. This was a way of keeping the structure from

remaining completely closed. In the recreation, we use Lindenmayer systems to recreate this ‘growth’ process. The L-systems operate on a grid similar to the original one as an intermediate step; spheres are used to represent the cells of the grid.

Each sphere (or cell) is linked to the 12 neighbours that surround it. For the recreation, we then use the following rule-replacements:

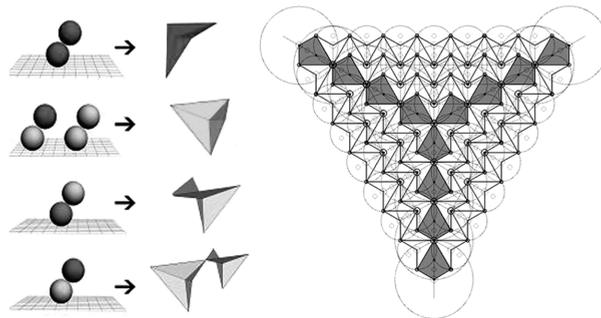


Figure 6. Rules for replacement and the structure after four iterations.

Figure 6 shows the structure after four iterations in the growth pattern. We now consider how this structure could be moved to a larger framework.

### 6. Using BIM to describe Reptiles

While this example was used for illustrative processes, there are many several lessons that can be gathered from this process. The most important being the coordination of the various units that evaluate code or scripts. The proposed architecture looks as follows:

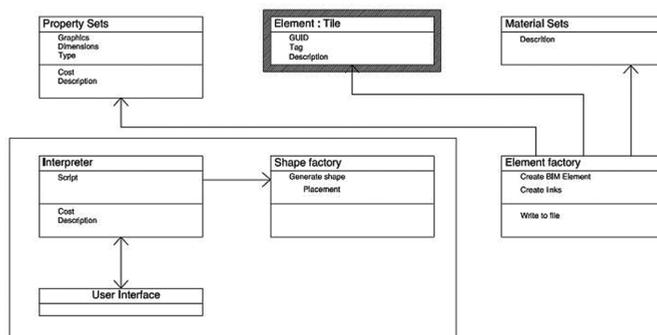
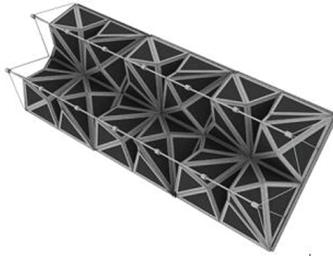


Figure 7. Having a tile produced by the element factory.

We introduce two conceptual units – the ‘Element factories’ - which help to automate the task of creating the object in the model with all the necessary references and the ‘Interpreter’ which can read an interpreted language. In this case, the language read is ‘C#’. The interpreter does not have to strictly follow the single thread of going from an input through a single process to an output, if concurrency is considered in the system architecture.. The following properties and representations shall be assigned to a tile: A cost, an outline which can be used for structural calculations and surface geometry for visualisation. Once an element has been established operations typical of parametric modelling can be performed on it.



*Figure 8. Truss system adapted from the base geometry of the ‘Reptiles’ tiles.*

## 6. Conclusion

Parametric models require more thought and planning than traditional models. They have two main advantages over static models. The first of these is the ability to make changes to the model rapidly and the second is to collect domain specific knowledge by creating assemblages that can be adapted to a given situation.

The approach to CAD using multiple unrelated files has been criticised by many as not being flexible to changes. It also means that the design process is likely to be a serial process instead of an iterative one. To be truly multi-disciplinary, a building information model is needed, as many of the disciplines outside of design, require meta-data and semantics to assess the model. This is because many disciplines require data beyond the geometry of the model – such as material and cost data. Some approaches such as Life-Cycle cost analysis require locations, so that the amount of energy embodied by the materials and their transportation can be calculated.

There is a currently a gap between the use of parametric and generative tools and modelling in practice. By generative design methods, we include work done with approaches as diverse as cellular automata, genetic algorithms, shape grammars, agent-based modelling etc. These techniques are

often deployed in explorative design. Allowing for parametric and generative methods within industrial standard tools can allow for a greater exploration of design. Including parametric modelling in a BIM framework however, brings with it some challenges as there is a need for more modules to handle the constraints, geometry building and input from the users. With generative design methods there is also a need for an interpreter to read the scripts that are used. In this paper, the 'Reptiles' project from 1968 was examined as it set the foundation for many generative experiments in architectural design. The aim was to utilise a cellular automata-like representation to fix local errors and create ensure only 'valid' models are created. Adjustments were made to how it might incorporate object data into its representation.

With regard to the second advantage mentioned of parametric modelling – the capturing of domain knowledge, it is an essential step that human designers work with higher, more expressive tools over time. One way in which this is possible, is by translating concepts from object-oriented modelling such as inheritance and encapsulation. By allowing this, a bridge can be made between the conceptual and developed stages of the design process, it also means that performance analysis can be conducted sooner and be part of the initial design explorations. As yet, there is much to be standardised with regard to parametric modelling in BIM.

## References

- Aish, R.: 2000, Migration from an individual to an enterprise computing model and its implications for AEC research, White paper submitted to the Berkeley-Stanford CE&M Workshop.
- Alexander, C.: 1977, *A Pattern Language. Towns, Buildings, Construction*, Oxford University Press, New York.
- Eastman, C., Teicholz, P., Sacks, R. and Liston, K.: 2008, *BIM Handbook: A Guide to Building, Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, Wiley, Hoboken.
- Flager, F., Somerekun, G., Welle, B., Haymaker, J. and Bansal, P.: 2008, *Multidisciplinary Process Integration and Design Optimization of a Relocatable Classroom Building*, CIFE Technical Report #TR175.
- Frazer, J.: 1974, Reptiles, *Architectural Design*, 4, 231–240.
- Gamma, E., Johnson, H. and Vlissides, J.: 1995, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Boston.
- Lee, G., Sacks, R. and Eastman, C. M.: 2006, Specifying parametric building object behaviour (BOB) for a building information modelling system, *Automation in Construction*, 15(6), 758–776.
- Pratt, M. and Kim, J.: 2006, Experience in the exchange of procedural shape models using ISO 10303 (STEP), in Martin, R. and Hu, S. M. (eds.), *Proc. 2006 ACM Symposium on Solid and Physical Modeling*, ACM, New York, 229–238.
- Stiny, G.: 2006, *Shape: Talking About Seeing and Doing*, MIT Press, Cambridge.
- Woodbury, R.: 2007, "Design Patterns". Available from: <http://www.designpatterns.ca>