Y. Ikeda, C. M. Herr, D. Holzer, S. Kaijima, M. J. Kim. M, A, Schnabel (eds.), *Emerging Experience in Past, Present and Future of Digital Architecture, Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA* 2015, 437–446.

PARAMETRIC BIM WORKFLOWS

PATRICK JANSSEN
National University of Singapore, Singapore
patrick@janssen.name

Abstract. Building Information Modelling systems enable the creation of associative parametric models that include sets of interlinked parametric objects. Graph-based modelling systems on the other hand enable the creation of parametric models with more complex iterative behaviours. Parametric BIM workflows aim to link graph-based systems to BIM systems. A key requirement of such workflows is the ability to generate associative BIM models. However, current approaches to creating such workflows are complicated by the fact that the process of cooking is only able to generate explicit geometry. An alternative approach is proposed in which the cooking process is able to generate associative models, thereby enabling more user friendly and streamlined BIM workflows to be created.

Keywords. Building Information Modelling, Parametric modelling, BIM workflows

1. Introduction

Parametric modelling (Woodbury, 2010; Janssen and Stouffs, 2015) and Building Information Modelling (BIM) (Eastman, 2008) are two modelling approaches that have recently gained much attention.

Complex BIM models can be very time consuming to build, and due to human error, mistakes and inaccuracies can accumulate. Furthermore, once the model is built, it may be difficult to make significant changes without rebuilding the model from scratch.

Parametric modelling can be divided into four main types: object modelling, associative modelling, dataflow modelling, and procedural modelling (Janssen and Stouffs, 2015). The key factor that differentiates these modelling types is how they support iteration. Object modelling does not allow for any iteration, associate modelling allows for single-operation iteration, data-

flow modelling allows for implicit multi-operation iteration, and finally procedural modelling allows for explicit multi-operation iteration. BIM systems only allow for associative modelling and as a result, they are limited in their ability to automate the model-building process.

Dataflow and procedural modelling on the other hand are much more powerful. To date, these modelling approaches have mainly been used to generate geometric models. However, it is clear that when BIM models with complex geometries need to be created, dataflow and procedural modelling approaches could be used to generate such models or parts of models. Such semi-automated generation of BIM models can be faster, less error-prone, and can enable more complex types of forms to be modelled. The automated process can generate components at various scales, from whole building models down to complex non-standard structural or façade details.

This paper focuses on the creation of loosely coupled parametric BIM workflows. Two types of parametric modelling will be discussed: associative modelling as used in BIM systems such as Autodesk Revit, Graphisoft ArchiCAD, and Bentley AecoSIM versus procedural and dataflow modelling as used in graph-based modelling systems such as Mc Neel Grasshopper, Bentley GenerativeComponents, Autodesk Dynamo, and Sidefx Houdini.

Section 2 investigates the limitations of existing BIM and graph-based systems with respect to parametric modelling. Section 3 analyses existing approaches to creating BIM workflows that allow graph-based systems and BIM systems to be linked. Section 4 proposes an alternative approach to creating parametric BIM workflows that overcomes fundamental limitations inherent in the existing approaches. Finally, section 5 briefly draws conclusions and indicates future avenues of research.

2. Limitations of BIM systems

In order to establish the need for new types of parametric BIM workflows, existing BIM and graph-based systems are first investigated.

A simple parametric modelling benchmark task is defined. The task has been crafted to highlight fundamentals limitations of the various BIM systems with regards to parametric modelling. The modelling task consists of generating the floor plates for a tapering tower whose overall form is defined by a curved surface. Floor plates for the tower are then generated at regular intervals, until the top of the tower is reached. The perimeter of the floor plates are associated with curved surface defining the tower form, thereby ensure that the floor plates automatically update themselves whenever the form is modified.

The total floor area of the tower is limited by the pre-define plot ratio for the site. The process of generating floors slabs therefore needs to start with the lowest floor and proceed upwards, keeping a tally of the floor area achieved up to that point. When this area reaches the required floor area, no more floors should be generated. This will ensure that the total floor area for the tower will not exceed the site plot ratio.

2.1. BIM SYSTEMS

The model cannot be built in the three BIM systems that were tested: ArchiCAD, Revit, and AecoSIM. The reason for this is that BIM system use an associative type of parametric modelling that only supports single-operation iteration. As a result, there is no way of representing the floor area constraint.

Of the three systems, Revit supports the most advanced types of associative relationships. In Revit, a model was built using a Conceptual Mass editing mode. The building form was defined as a conceptual mass and when it was modified, the floor slabs automatically updated. However, these underlying associative mechanisms do not support the type of multi-operation iteration required for representing the floor area constraint.

2.2. GRAPH-BASED SYSTEMS

Although the benchmark task cannot be modelled in the BIM systems, it can easily be modelled in any of the graph-based systems. The task thereby underlines the need for parametric BIM workflows.

For dataflow modelling, the benchmark model was built in both GenerativeComponents and Grasshopper. These systems only support implicit multi-operation iteration. This means that the iterative behaviour is achieved using custom data structures in combination with data matching algorithms that appropriately interpret these data structures. Due to the implicit nature of the iterative process, two steps are required. In the first step, all the floor plates are generated without taking into account the floor area limit. In the second step, the areas of each floor plate are calculated and those that result in the floor area limit being exceeded are then deleted.

For procedural modelling, the benchmark model was built in both Houdini and Dynamo. In this case, explicit multi-operation iteration is supported. In Houdini, this was achieved using ‘for each’ nodes combined with data sinks, while in Dynamo it was achieved using recursion. In both cases, the iterative loop is explicitly represented, thereby avoiding the need for a two-step process. In the iterative loop, each floor plate is generated and a tally is maintained of the total floor area achieved up to that point. As soon as the maximum floor area is exceeded, the iterative loop is exited.

3. Linking graph-based systems and BIM systems

In order to be able to combine dataflow or procedural modelling with BIM modelling, two approaches are possible: the embedded approach and the coupled approach. With the embedded approach, BIM systems are extended by adding support for dataflow or procedural modelling. With the coupled approach, dedicated graph-based systems are coupled to BIM systems, thereby allowing graph-based systems to be used to generate models, and BIM systems to manage the data.

A fundamental challenge with the embedded approach is the fact that the core use cases of graph-based systems and BIM systems are highly divergent. Graph-based systems focus on design exploration, requiring light-weight minimal models that are responsive and quick to update. BIM systems focus on design interrogation, requiring maximal models that incorporate detailed information that can be queried.

Creating a single BIM system that is adept at supporting both exploration and interrogation may not be viable for two reasons. First, BIM models are by their very nature large complex datasets. As a result, allowing users to parametrically explore such models may severely reduce the latency and robustness of the system. This is already evident within Revit, where making parametric constraint-based changes to large models can become slow and may often result in errors whose sources are unclear even to expert users. Second, BIM systems already have very complex user interfaces and adding advanced dataflow and procedural modelling capabilities may result in a user-interface that is overly complex for either use case. Again taking Revit as the example, the user interface can already be seen to be very complex, with multiple different but interrelated modelling modes, resulting in a steep learning curve for novice users.

This paper therefore argues that the coupled approaches that link graph-based systems and BIM systems is preferable. These workflows are referred to as parametric BIM workflows.

3.1. PARAMETRIC BIM WORKFLOWS

A range of tools has started to emerge that support parametric BIM workflows using either a tightly coupled approach or a loosely coupled approach (see Figure 1).

With the tightly coupled approach, systems are coupled through the Application Programming Interface (API) provided by the BIM system. In this case, graph-based systems communicate via the API of the BIM system, directly instantiating geometry in the BIM model each time the graph-based model is executed. Examples of this approach are GenerativeComponents

which uses the AecoSIM API, and Dynamo, which uses the Revit API. (Note that both GenerativeComponents and Dynamo can also be used within loosely coupled approaches.)

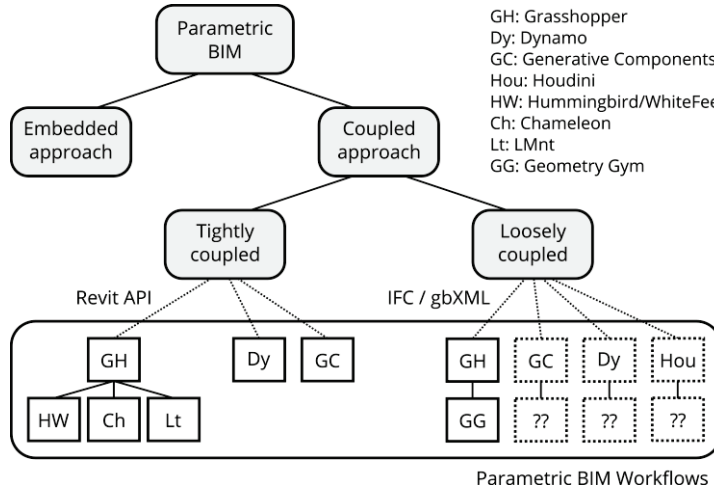

Figure 1: A hierarchy of different approaches to combining graph-based systems and BIM systems.

With the loosely coupled approach, systems are coupled through model exchange. The graph-based system typically generates data in a standard file format that can be directly imported into the BIM system. An example of this approach is Grasshopper/GeometryGym, using IFC as the exchange format, and Grasshopper/Chameleon using gbXML as the exchange format.

These approaches are still evolving. However, of the two approaches, the loosely couple approach using file exchange has the fundamental advantage that it is workflow agnostic, allowing users to link together tools and systems to support various forms of collaboration and exchange. For example, since GeometryGym outputs a standard IFC file, users have the choice to link to any BIM application that can import an IFC file. (In practice, there are still many issues with IFC interoperability. However, over time, it is likely that these will be resolved.)

Figure 2 shows the various different models involved in a loosely coupled parametric BIM workflow linking graph-based systems and BIM systems. The first model is the parametric model, which is either a dataflow or procedural graph. Given a set of inputs, this model can be used to generate the second model through a process that is referred to as ‘cooking’. The cooked model can then be used to generate the third model, which is the ex-

change model in a format such as IFC. Finally, the BIM system can be used to create the fourth model, by importing the exchange file.

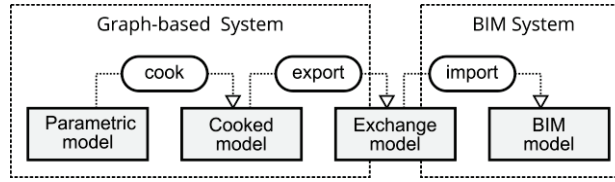

Figure 2: Loose coupling between graph-based systems and BIM systems.

3.2. EXPORTING ASSOCIATIVE EXCHANGE MODELS

The target BIM model is not an explicit geometric model, but rather is an associative model that incorporates interlinked parametric objects. If the exporter is only capable of generating explicit geometric models, then all the parametric and associative intelligence within the model will be lost. This would undermine many of the benefits in using BIM in the first place. As a result, exporters must be capable of exporting associative models.

Formats such as IFC already incorporate the ability to represent interlinked parametric objects. For example, an IFC wall can be represented in a number of different ways (IFC4, 2014). The 'boundary representation' is a non-parametric representation that explicitly defines all the bounding surfaces of the solid wall object. In addition, a number of parametric representations are possible. The 'swept solid' representation defines the solid wall object by sweeping a planar profile using either linear extrusion or revolution techniques. The 'clipping' representation defines the solid wall object as a result of series of Boolean operations in a Constructive Solid Geometry tree. Finally, the 'standard case' wall type is a parametric object that defines a wall by the centre line and parameters that specify the width and height. Furthermore, walls can have associative relationships with other objects that affect their final form.

In order to be able to create parametric BIM workflows, exporters must be capable of transforming the cooked model into an associative exchange model. However, developing such exporters is very difficult due to the fact that the cooked model contains only explicit geometry.

Figure 3 shows the various model types at each stage of the workflow. In this example, a dataflow model generates two walls that then need to be exported to an exchange file using the 'standard case' representation based on the wall centre line. Cooking the dataflow model results in explicit geometry consisting of a total of 12 polygon faces. Creating an exporter capable of

mapping this set of polygons to the ‘standard case’ wall representation is very complex, even for such a simple case.

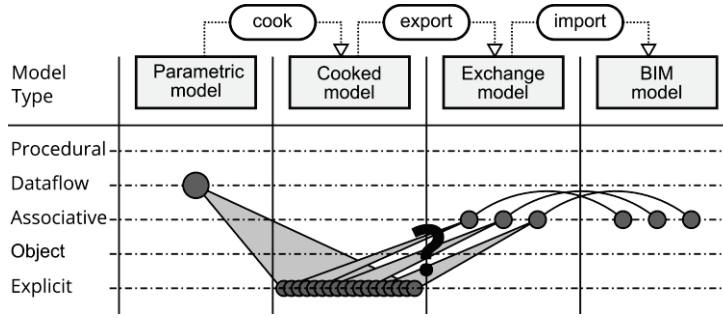

Figure 3: An impractical strategy for parametric BIM workflow.

Due to the difficulty in creating such exporters, existing plugins have developed various workarounds. One approach is to use the dataflow model to directly generate the exchange model, thereby avoiding the need for an exporter altogether. This is the approach taken by the GeometryGym plugin. The advantage is that it does not need to deal with the explicit geometry generated by the cooking process. However, the disadvantage is that the dataflow model becomes riddled with data exchange nodes that have very marginal relevance to the parametric modelling task. These additional nodes significantly increase the complexity of the dataflow graph.

4. Proposed strategy

A strategy is proposed for creating parametric BIM workflows that avoids explicit geometry and at the same time maintains a clear separation between the process or creating parametric models and the process of generating exchange files.

The proposed strategy is shown in Figure 4. The example being considered is the same as the one depicted in Figure 3, where a dataflow model is being used to generate two walls. The key difference in this case is the fact that the cooking of the dataflow model does not result in explicit geometry but instead generates a set of associative objects. This significantly simplifies the development of exporters since the cooked model and the exchange model are now both associative models.

In order to realize such workflows, new types of graph-based systems need to be developed. The main issue relates back to iteration. With dataflow or procedural modelling, the user can define an iterative loop where a set of operations are repeatedly executed. Currently, graph-based systems can use this type of iteration to generate varying geometric structures, each explicitly

represented. With the proposed strategy, the user would be able to generate the same varying geometric structures, but instead of using an explicit representation, each structure would be represented as an associative object that retains the specific operations used to generate it. The proposed object-based associative representation consists of objects defined by some root geometry that is then modified by a sequence of actions, called 'modifiers'.

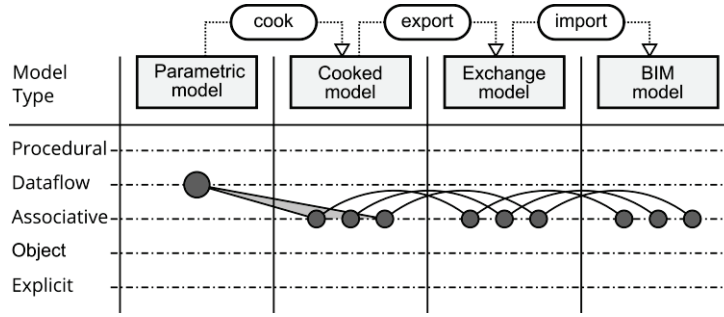

Figure 4: The proposed strategy for parametric BIM workflows.

The proposed object-based associative representation is similar to the representations used in other existing system that support associative modelling (Janssen and Stouffs, 2015). Scene-based systems such as Autodesk 3DS Max are used to create scenes populated with objects, with each object being generated using sequences of operations called 'modifiers'. Feature-based systems such as Autodesk Inventor are used to create assemblies of parts, with each part being generated by a sequence of operations called 'features'.

4.1. AN EXAMPLE

Figure 5 shows an example of a dataflow model, and compares the explicit and associative representations. This example depicts a dataflow graph that uses a curved line to define the position of two walls.

On the left side, an abstracted view of a dataflow graph is shown. Modelling operations are depicted as ellipses and data sets are depicted as boxes (Janssen and Stouffs, 2015). The graph consists of five operations: 1) a curve is created from a set of three points, 2) the curve is divided into 5 line segments, 3) three of the line segments are deleted, 4) the remaining two line segments are extruded to make faces, and 5) the faces are thickened to make solids. Each operation has a set of parameters labelled *p1* to *p5* respectively. The geometry that is generated by each node is shown next to the dataflow graph.

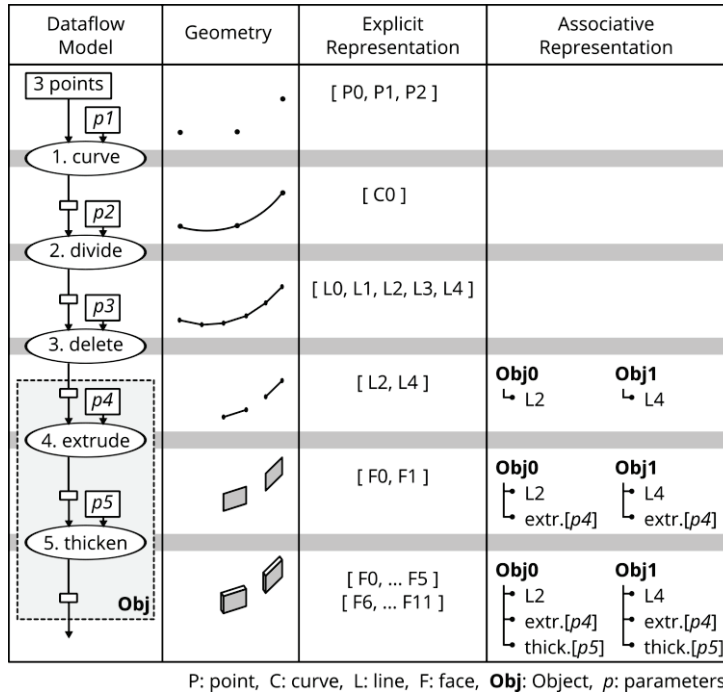

Figure 5: A comparison of the explicit representation versus the associative representation.

The explicit data representation is based on storing a set of geometric entities in a simple list-based data-structure. In the example shown in Figure 5, the process starts with a list of 3 points. Each operation consumes and produces lists of data. The 'Explicit Representation' column shows the explicit data for each operation. The 'curve' operation produces a list containing a single curve. The 'divide' operation produces a list of 5 line segments. The 'delete' operation produces another list of lines, in this case containing just two lines. The 'extrude' operation iterates over the lines list one at a time, producing a list of two faces. Finally, the 'thicken' operation iterates over the faces list, and produces two lists of 6 faces each.

In Figure 5, the dashed box on the left labelled 'Obj' signifies that associative objects needs to be generated. The 'Associative Representation' column shows the associative data for each operation. The first three operations do not have any associative data. Only when the dataflow enters the 'Obj' box is the generation of associative objects triggered. The two lines generated by the 'delete' operation are converted into two objects, with the root entity for each object being the explicit representation of the line. The 'extrude' operation then iterates over these objects and appends an 'extrude' modifier

to each object, along with the extrude parameters. Finally, the 'thicken' operator takes the two extruded objects and appends a 'thicken' modifier to each object, along with the thicken parameters.

The advantage of the object-based associative representation is that each object can later be interrogated in order to reveal its construction history. This renders the task of creating BIM model exporters much more straightforward. For example, if the two walls needed to be exported using the 'standard case' wall representation, then the centre-line and the width and height parameters could easily be extracted from the objects. Furthermore, other representations such as the 'swept solid' or 'clipped' representation could also be easily derived.

In general, the development of graph-based systems with object-based associative representations has the potential to create more user-friendly and streamlined parametric BIM workflows.

5. Conclusions

The linking of graph-based systems and BIM systems into parametric BIM workflows enables the semi-automated generation of models with complex geometries. However, current approaches to creating such workflows are complicated by the fact that the process of cooking dataflow and procedural models is only capable of generating explicit geometry.

An alternative approach has been proposed that results in more user-friendly and streamlined parametric BIM workflows. This approach requires the development of graph-based systems that use an object-based associative representation.

Future work will focus on the development and implementation of a prototype procedural modelling system that supports the proposed object-based associative representation (Janssen, 2014). The prototype will allow experiments to be conducted in creating and evaluating parametric BIM workflows.

References

Eastman, C.: 2008, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, Wiley.

Janssen, P.: 2014, Visual Dataflow Modelling: Some thoughts on complexity, *Proceedings of the 32nd eCAADe Conference*, Newcastle, UK, 547–556.

Janssen, P. and Stouffs, R.: 2015, Types of Parametric Modelling, *Proceedings of the 20th International Conference on Computer-Aided Architectural Design Research in Asia* (CAADRIA 2015), to appear.

Woodbury, R.: 2010, *Elements of Parametric Design*, Routledge.

IFC4 2014, http://www.buildingsmart-tech.org/ifc/IFC4/final/html/