R.Doe, M.Aitchison, *Facilitating change: the modular format in the design of prefabricated homes: 50th International Conference of the Architectural Science Association 2016,* pp. 1–10.

Facilitating change: the modular format in the design of prefabricated homes

Robert Doe, Mathew Aitchison
The University of Sydney, Sydney, Australia
rdoe8535@uni.sydney.edu.au, m.aitchison@sydney.edu.au

Abstract: This paper examines the modular format as a tool for systematising both computational design rules and building assembly methods. Its focus is a live case study of a 3 storey prefabricated apartment block. The goal is to explore the utility and effectiveness of this tool and its ability to facilitate change. A parametric model of the apartment block is made and various aspects of the modular format are applied to its assembly and components. Combined as a computational design and building assembly tool, the benefits of the modular format may be exploited. Thus flexibility, adaptability, changeability and reusability are maximised. Whilst fully interactive integration of the modular format may not yet be possible, it is hoped that its incorporation into the complexities of the architectural design process may at least result in more authentic outcomes.

Keywords: modular format, design rules, parametric modelling, prefabrication

1. Introduction

Architects find the implementation and conceptual structure of computational design methods challenging. More easily understood and flexible expressions of design rules are required to stimulate engagement with its tools, and it has been urged that these rules should be expressed in ‘modular’ format (Mitchell, 1989). Although this reference is made to the design rule as computational expression of a procedure delineating design requirements as input and the result as output, the notion of the modular format assists with comprehension of all complex artificial systems, including building assembly processes. A modular format denotes discrete but connectable components, well-defined interfaces, reusability and shareability, in contrast to integrated formats which have no clear division between components. This paper provides understanding and demonstrates implementation of the modular format and its ability to accommodate change, a ubiquitous feature in the practice of architecture. The modular format as expression of a computational design rule and as a means of building assembly is examined in relation to a case study of a live architectural design project.

2. Methodology

The AP14 project is a 3 storey 26 apartment development in Brisbane, Australia. It is both a commercial development designed by Fairweather Jemmott Architects and a research project, funded by the Queensland Government Accelerate Partnerships program and designed by Aquatonic, to compare prefabricated assembly with traditional building methods (Figure 1). A hybrid prefabrication strategy proposed by the research team was confirmed by Bligh Tanner's structural engineers. Thus, the building assembly's modular format might be described as the integration of hybrid components - panellised floor, wall and roof elements, volumetric bathrooms pods, and clip-on balcony components.

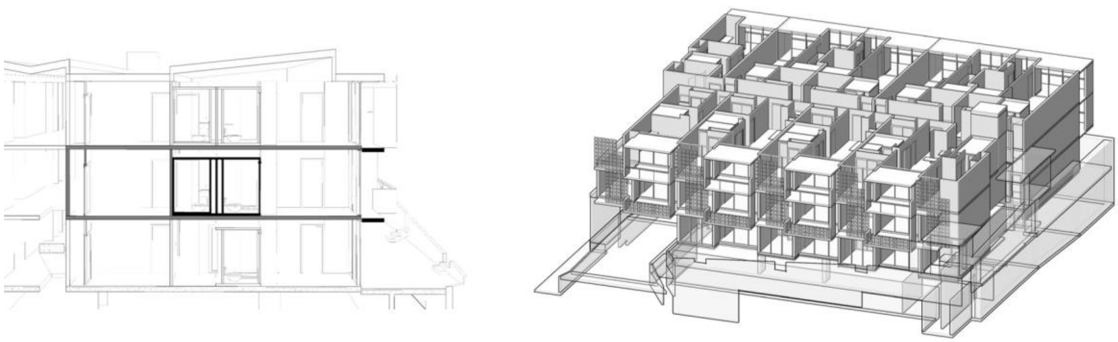

Figure 1 AP14: unit type in section & whole scheme in axonometric

The study uses parametric modelling tools to explore expression of design rules in modular format: a 'component based modeller' (e.g. Autodesk Revit, ArchiCAD and Microstation); a 'design development program' (e.g. CATIA, PTC's Creo and Solidworks) (Schodek *et al.*, 2005); and a visual data flow graph (e.g. Grasshopper, Autodesk Dynamo) to further examine the design rules in both modular and declarative formats. The modular format has guided the creation and implementation of discrete but connectable components, well-defined interfaces, reusability and shareability. Its intelligibility, ease of use, flexibility and adaptability to change will also be discussed.

3. The modular format

The key characteristics of the modular format are defined above, however more generally it may be described as the decomposition of a system into parts as a means to understand and manage complexity. Furthermore, the concept of connectability by means of well-defined interfaces acknowledges the dynamic nature of interrelationships between these parts.

The modular format as a computational expression dealing with complexity has developed from discourse in other fields. For example, von Bertalanffy's General Systems Theory (von Bertalanffy, 1950) captured a growing tendency in the science of its time to recognise the importance of dynamic interaction in all fields of reality, marking a shift from a mechanistic and static understanding of systems to one recognising their organic and dynamic tendencies. Simon (Simon, 1969), whilst disagreeing with the abstraction of these characteristics of complex systems to general principles, reiterated the need for hierarchy and the decomposition of complex systems into functional parts. These insights are useful in consideration of the nature of all complex artificial systems, including computational design and building assemblies. But the modular format has been defined in alternative ways in the computational design and building assembly fields. There follows a brief account of these definitions, followed by a description of a strategy which utilises and integrates the modular format in these fields.

Rudolph Schindler in the early twentieth century, proposed use of a modular unit as a regulator of architectural proportion and building assembly, and indeed, Le Corbusier's Le Modulor was a sub-set of Schindler's 'universal set' of modular ratios (March, 2003). These modular units represent early development of computational methods to manage the modular format and significantly, the attempt to mentally compute proportional and scalable modules. Schindler firmly believed that the modular unit was a 'reference frame in space', the most important mental computational tool in the architect's toolbox (Park, 2005). His proposition presaged Christopher Alexander's work on patterns, which was more characteristic of a modular format as it described discrete connectable components, but with a similar conviction that computation is a mental process dependent on the understanding and integration of subdivisions of the whole (Alexander, 1964; 1979).

Although a major flaw in Alexander's work was a mechanistic assumption that a set of design requirements can be listed at conceptual design stage (Lawson, 2005), nevertheless interactions between these descriptions of patterns and computer programmers of the time became the basis for a branch of object-oriented programming (Lea, 1994). Furthermore, software engineers developed structured programming in the 1960's, focused in particular on the benefits of the modular format, a strategy adopted as a response to increased complexity in computer program flows (Davis, 2013). Computational designers, in turn, absorbed lessons from software engineers. In 1963 Sketchpad (Sutherland, 1963) introduced parametric modelling to the CAD world, enabling dynamic interrelationships between elements to be represented, and their manipulation by 'light-pen' and keyboard. As techniques of parametric modelling have developed the modular format has been proposed as a method for ensuring models remain disentangled, thus it should be expressed as self-contained chunks of code, shareable, reusable and debuggable (Davis *et al.*, 2011).

Building assemblers, by the mid-twentieth century, had developed their own definition of the modular format, which found expression as a volumetric form which endures today (Garrison and Tweedle, 2008). In the USA it is epitomized by 'transportable' or 'manufactured' homes which are shipped to site on a flat-bed truck, such that 2-4 modules may be joined together to form a whole house (Steinhardt *et al.*, 2013). These possess some characteristics of the modular format - discrete envelope, reusability - but they are essentially mechanistic and static representations with limited consideration of interface connections between modules. Encouraged by observation of the manufacturing industry's products these modular homes are assembled from standardised parts at the end of a chain of production, thus facilitating limited potential for customisation (Carpo, 2011).

However, this volumetric form is only one interpretation of the modular format in the field of building assembly. A Swedish study (Jensen *et al.*, 2012) takes this discussion further by focusing on an industrialised building system and the customisation of its floor components, which were enabled by their modular format and configuration. In this study it was the interface between modules, the enabler of connections - 'slot', 'bus', 'sectional' - which more meaningfully defined the nature of their modular format. Furthermore, these interfaces facilitated sharing and reuse, characteristics common to modular formats in the computational design field. This prompts consideration of strategies which make best use of the modular format in both fields

An effective strategy in the design process, which utilises the modular format, is to move beyond high design content with its extravagant outcomes to a low-design content, non-repetitive design strategy, "… an elegantly minimal, tightly disciplined response" (Mitchell, 2005). For example, 'Kilian's roof' is a computational design exercise which has successfully demonstrated this strategy as follows: a framework is defined to lead sub-division of the complex whole; then a point collection is created to assist with location of a proxy object; finally module components are associated with the proxy object (Woodbury *et al.*, 2007). A built example of this strategy of low design, in-context complexity, incorporating non-repetitively designed modules is the British Museum's Great Court roof, London, UK designed by Foster and Partners, in which the varied shape of roof panels and structural members is controlled by simple rules and parameter values (Figure 2).

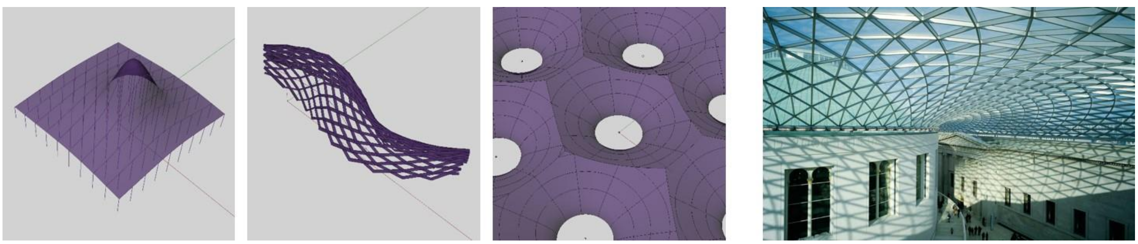

Figure 2 'Kilian's roof' design exercise (image source: Daniel Davis's PhD thesis, RMIT 2013) &, the Great Court roof, British Museum (photo copyright: Ben Johnson).

4. Case study – AP14

Aspects of the modular format - discrete but connectable components, well-defined interfaces, reusability and shareability - are examined in the context of the AP14 project, focusing on the design of panellised wall components and the building's assembly. Incorporation of the strategy outlined above - low-design content, non-repetitive design - is also considered.

4.1 Modular format wall panels

As noted above, Schindler's approach pre-dated modular computational programming and found a way to maximise mental computational ability by defining the modular unit as a subdivision of the whole which could be held in mind throughout the design process. However, computational parametric ability made the design of proportional and scalable relationships a simple and perhaps reductive exercise. Nevertheless, his concept of a 'reference frame in space' echoes the process by which a parametric model is set-up. With parametric modelling, Schindler's reference frame becomes a reference plane to which geometry is aligned. This has been described elsewhere as set-up of the 'design domain' (Burry, 2007) with which, by association, geometry updates as dimensional parameters change. The modular format may thus find expression in the AP14 project as a timber engineered wall panel suitable for all situations predicted (Figure 3).

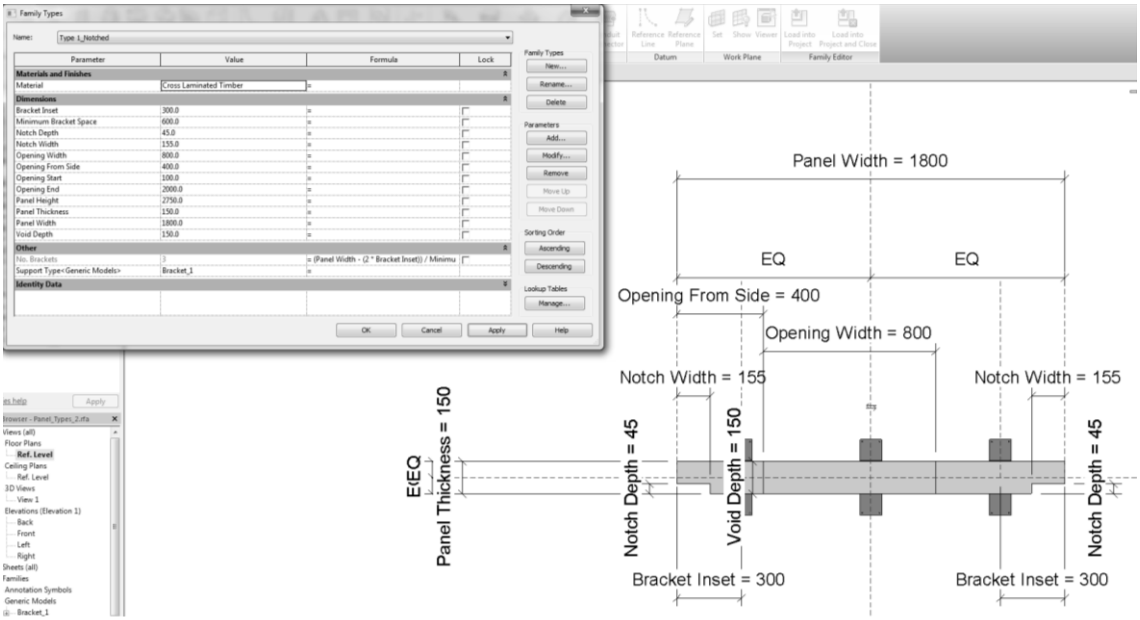

Figure 3 Modular format wall panel: parameters, rules & reference planes (Autodesk Revit)

This is an example of the strategy of low-complexity, non-repetitive design as this panel may be instantiated as many different types to suit material and engineering requirements so that it might be notched, un-notched, solid, have openings, and have bracket fixings automatically located (Figure 4). These panels are then loaded into the unit's group for scheduling and creation of shop drawings, a method which is extensible, flexible and able to accommodate change.

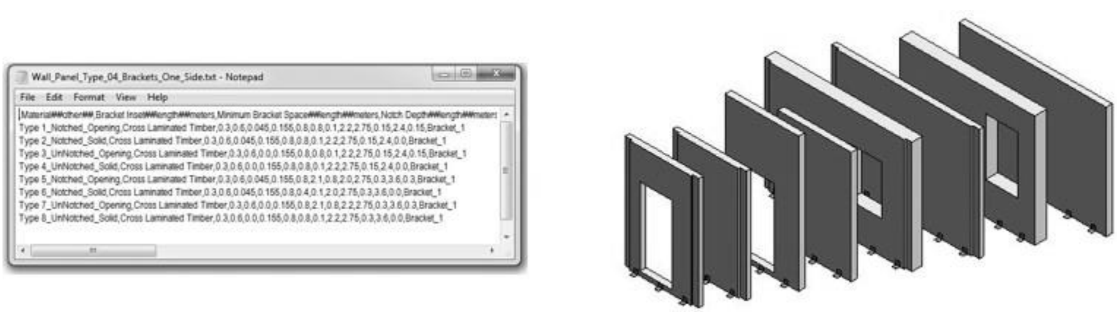

Figure 4 Modular format wall panels: type catalogue, types & instancies (Autodesk Revit)

4.2 Modular and declarative format programs

Mitchell (Mitchell, 1989) urged use of modular and declarative formats to express computational design rules to make them more easily understood and more easily modifiable. The benefits of this combination have been demonstrated in structuring parametric models and include, flexibility in accommodating change and the ability to defer design decisions to later stages (Aish and Woodbury, 2005; Davis, 2013). In this exercise using a visual data flow graph (Autodesk Dynamo), the modular format is illustrated on the left with clear naming, identification of inputs and outputs, operations encapsulated in nodes, plus a description of its purpose (Figure 5). The declarative format is also illustrated on the left by its wires and nodes describing what is wanted but not how it is achieved, in contrast to imperative formats which describe in script form each step to achieving the desired outcome. Separation with the instance illustrated on the right facilitates deferral of design decisions.

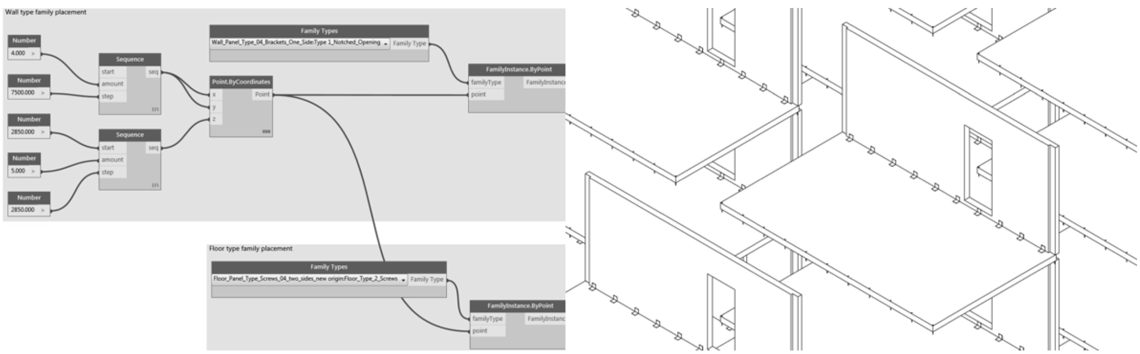

Figure 5 Modular & declarative formats defining wall & floor panels (Autodesk Dynamo)

The associational and logical relationships represented by this modular and declarative format certainly assist with understanding of the design rule – the procedure stating design requirements as inputs and outputs. However, this is a relatively simple design task and recently it has been argued that the declarative format in particular does not scale to more complex logic, and therefore to more complex building assembly models. One solution to this problem may be to combine declarative visualisation with imperative programming incorporating iterative looping and higher order functions (Aish, 2013; Janssen *et al.*, 2016).

4.3 Modular format building assembly

This next exercise focuses on setting up several flexible models to reduce design complexity by incorporating pre-assembled components. It has been demonstrated, in an experiment using a design development program (CATIA) to test the flexibility of a parametric model whilst structural elements were being optimised, that set-up of an "… all-encompassing flexible model… is not advisable", because parameters affected by optimisation interfere with the hierarchical and logical structure of the model (Holzer, Burry & Hough 2007). It has also been demonstrated, in another computational design test of a visual data flow graph's ability to design 'Kilian's roof', that several modular reusable nodes are preferable to a single complex node (Janssen 2016). Both these examples suggest that set-up of computational design rules should follow a modular format.

However, reduction of the design complexity of the building assembly should also be considered. This can be achieved by use of prefabricated or pre-assembled elements in which a degree of design content is inherited (Mitchell 2005). It has also been proposed, with reference to assemblies of components and identifiable sub-systems of buildings, that 'composition' as a computational design tool should be 'internalised and operationalised' by designers (Aish, 2005) – a description which might further be seen as an echo of Schindler's and Alexander's earlier ambitions. Thus the modular format of the building assembly may be more closely interpreted by the modular format of the computational design set-up to reduce design complexity overall and improve flexibility and adaptability to change.

As noted previously, in the AP14 project the prefabricated elements are engineered timber floor, wall and roof panels, whilst the pre-assembled elements are bathroom pods and clip-on balconies (Figure 6).

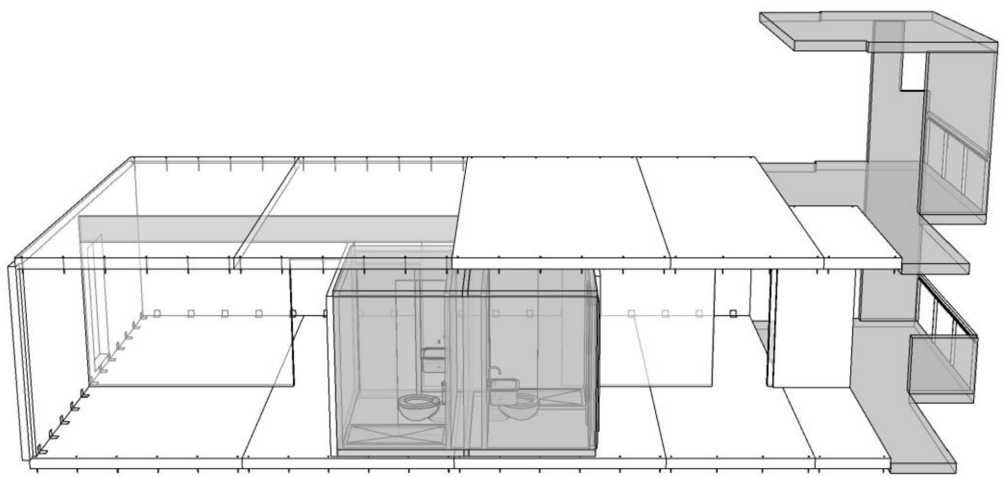

Figure 6 Modular format building assembly (Autodesk Revit)

The modular format is defined by these elements - self-contained with connectable interfaces, reusable, shareable, - and their relationship or assembly. A component based modeller (Autodesk Revit) is used to assemble a unit type's elements: thus combining these elements into a 'group' clarifies interface connections whilst maintaining their discrete properties; 'group' amendments are non-repetitive as a change to one instance propagates to all; a 'group' may also be linked to the main model and amended in isolation by another design team member thus enhancing shareability. The unit type 'group' itself is constrained to the vertical (grids) and horizontal planes (levels) of the main model's framework. If a grid or level is changed the unit and its 'group' geometry changes (Figure 7).

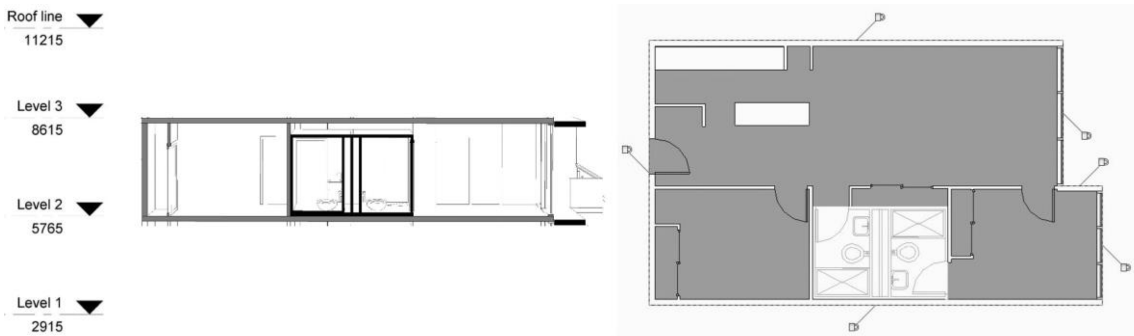

Figure 7 Unit type levels associated with main model & unit type constraints (Autodesk Revit)

It should be noted that with component based modellers, the terms ‘part’ and ‘assembly’ may have a meaning different to common usage in the manufacturing field. Therefore a ‘part’ may be a wall layer in a ‘system family wall’ (Autodesk Revit), rather than a discrete, connectable modular entity. Whilst ‘assembly’ (Autodesk Revit) may simply refer to the arrangement of elements into a 2D shop drawing. Component based modellers’ focus on building information modelling (BIM), organising information for availability over the life of the building, whilst the geometry of the structure and components - the realm of computational design - often remains inaccessible to the architect.

In contrast, a design development program with full parametric capabilities (Digital Project/CATIA), deals with parts and assemblies in a hierarchical and logical manner, in a way familiar to the manufacturing industry. Its hierarchical tree represents the organisation of the model: a framework sketch defines grids and levels; and parts are assembled within a ‘product’ file as illustrated here (Figure 8).

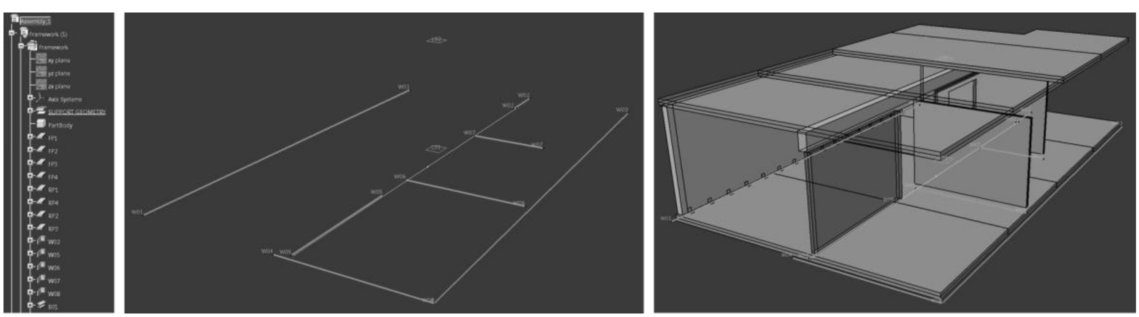

Figure 8 Hierarchical tree, framework sketch & product assembly (Digital Project / CATIA)

Thus, the modular format may be more logically represented using design development programs compared with component based modellers. However more discipline is needed to set-up its hierarchy of frameworks, parts and assemblies and more cognitive effort is needed to define parametric relations and constraints between geometrical elements.

5. Conclusion

In summary, architects are keen to utilise computers to their best advantage, and they want their tools to be intelligible, flexible and adaptable if they are to cope with change amid the dynamics of the design process. They are also aware of, and eager to harness, the direct link between these new design tools and better ways of fabricating and assembling buildings.

The results of these experiments confirm significant implications for integration of the modular format in the fields of computational design and building assembly. Modular formatted wall panels demonstrate that the strategy of low-complexity, non-repetitive design is extensible, flexible and able to accommodate change. However, more research is needed to explore the advantages such a strategy might have when focused on other modular and prefabricated components. It is also important to further examine the nature of self-contained, connectable interfaces, and their ability to be reusable and shareable.

It has also been shown that modular and declarative formatted programs improve intelligibility and ease of use. But more testing is needed to examine the capability of declarative formats in dealing with greater complexity, and to determine the extent to which the imperative format should be integrated with these other formats.

Examination of modular formatted building assemblies also highlighted synergies with modular formatted computational design tools. Furthermore, it was shown that design complexity can be further reduced by incorporation of pre-assembled and prefabricated components thus improving flexibility and adaptability to change. If design development programs are also utilised then representation of the modular format is enhanced because of their inherently hierarchical set-up and ability to deal with complex logical relationships.

This paper has interpreted the modular format as an expression of design rules in computational design and as a feature of building assembly models. The modular format facilitates change by systematising the complexity of the design and assembly processes and, far from being a static or mechanistic concept, the modular format combined with parametric modelling tools becomes part of a dynamically controllable association of parts to the whole. If computational design methods and building assembly strategies adopt the concept of the modular format then, subject to the further research identified above, its full benefits might be exploited: flexibility, changeability, reusability, and shareability.

To be worthwhile and effective the modular format as a computational design tool should also enable the architect to progress fluently from concept design to fabrication-ready information, from simple to complex building models. And it should also be acknowledged that increasingly, rather than traditional documentation outputs, minimal geometrical information may be needed for fabrication-ready information. Therefore, the modular format should not only operate smoothly from concept to fabrication but should also reflect this new form of design which is 'open-ended' (Aish, 2013), and therefore in harmony with current tendencies favouring flexibility, adaptability and participation.

A corollary of incorporation of the modular format across architecture, not just in its design tools, might be a better response to the demands of complex conditions. Rather than the "seduction of the surprising" it might give rise to a more, "authentic architecture of the digital age" (Mitchell, 2005).

Acknowledgements

Daniel Davis, thanks for permission to use Figure 2 ('Kilian's roof'), PhD thesis, RMIT 2013.
Ben Johnson (copyright), photo 1705118/img3, Foster & Partners, Great Court roof, British Museum, UK.

References

Aish, R. (2005) From Intuition to Precision, *eCAADe23*, Lisbon, Portugal.

Aish, R. (2013) DesignScript: scalable tools for design computation, *eCAADe31*, Delft, Netherlands.

Aish, R. and Woodbury, R. (2005) Multi-level interaction in parametric design, *Smart Graphics, 5th International Symposium*, Germany, pp. 151–162.

Alexander, C. (1964) *Notes on the synthesis of form*, ed., Harvard University Press, Cambridge, MA, USA.

Alexander, C. (1979) *The Timeless Way of Building*, ed., Oxford University Press, New York.

Burry, J. (2007) Mindful spaces: computational geometry and the conceptual spaces in which designers operate, *International Journal of Architectural Computing*, 5(4), 611-624.

Carpo, M. (2011) *The Alphabet and the Algorithm*, ed., MIT Press, Cambridge, MA, USA.

Davis, D. (2013) *Modelled on software engineering: flexible parametric models in the practice of architecture*, PhD dissertation, School of Architecture and Design, RMIT University, Melbourne, VIC, Australia.

Davis, D., Burry, J. and Burry, M. (2011) Untangling Parametric Schemata: Enhancing Collaboration through Modular Programming, *CAAD Futures 2011 - Designing Together*, Liège, Belgium.

Garrison, J. and Tweedle, A. (2008) *Modular Architecture Manual*, ed., Kullman Buildings Corporation, USA.

Janssen, P., Ruize, L. and Mohanty, A. (2016) Mobius: a parametric modeller for the web, *CAADRIA*, Hong Kong.

Jensen, P., Olofsson, T. and Johnsson, H. (2012) Configuration through the parameterisation of building components, *Automation in Construction*, 23, 1-8.

Lawson, B. (2005) *How Designers Think - The Design Process Demystified*, 4th ed., Architectural Press, UK.

Lea, D. (1994) Christopher Alexander: an introduction for object-oriented designers., *Software Engineering Notes*, 19(1).

March, L. (2003) Rudolph Schindler: space reference frame, modular coordination and the 'row', *Nexus Network Journal*, 5(2), 48-59.

Mitchell, W. J. (1989) A new agenda for computer-aided design., *CAAD Futures*, Vienna, Austria.

Mitchell, W. J. (2005) Constructing Complexity, *CAAD Futures*, Vienna, Austria.

Park, J.-H. (2005) Rudolph M. Schindler: proportion, scale and 'row', *Nexus Network Journal*, 5(2), 60-72.

Schodek, D., Bechthold, M., Griggs, K., Kao, K. M. and Steinberg, M. (2005) *Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design*, ed., John Wiley & Sons Inc., NJ, USA.

Simon, H. A. (1969) *The Sciences of the Artificial*, ed., MIT Press, Cambridge, MA, USA.

Steinhardt, D. A., Manley, K. and Miller, W. F. (2013) *Profiling the nature and context of the Australian prefabricated housing industry*, Queensland University of Technology, QLD, Australia.

Sutherland, I. E. (1963) *Sketchpad: a man-machine graphical communication system*, PhD dissertation, MIT, MA, USA.

von Bertalanffy, L. (1950) An outline of General System Theory, *British Journal of the Philosophy of Science*, 1(2), 134-165.

Woodbury, R., Aish, R. and Kilian, A. (2007) Some patterns for parametric modelling, *ACADIA*, Nova Scotia, Canada.