(Re)design of construction assemblies with function/behaviour/structure grammars

CHASE Scott C.

University of Strathclyde
http://homepages.strath.ac.uk/~cas01101

A formal framework for redesign based upon Function/Behaviour/Structure models and design grammars is described. A proposed application domain is for the design and redesign of construction assemblies. GDL object technology is proposed as a candidate tool for implementation.

***Keywords**: Redesign; design grammars; Function/Behaviour/Structure models; construction components; Geometric Description Language*

Background

This paper describes a framework for the design and redesign of construction assemblies, building upon previous work with function/-behaviour/structure grammars. (Chase and Liew, 2001). There have been a number of generative systems developed for detail or component based design, but they usually entail the use of an expert system or other reasoning mechanism for the assembly (e.g. Beesley and Seebohm, 2000; Harfmann, 1993). Another possible paradigm for detail design has been parametric variation; the Topdown system allowed simultaneous top down and bottom up design processes (Liggett et al., 1990). Design grammars in general have not been utilised for construction detailing; one notable exception is the EAVE system for roof details (Mitchell and Radford, 1987). Function based design grammars are becoming commonplace in engineering domains (e.g. Starling and Shea, 2002). Here we utilise a grammar based methodology in combination with a function/behaviour/structure representation, but as a facilitator of detail redesign by substitution.

Framework for redesign

The framework described here builds upon the original framework, which was predicated upon redesign only. In the earlier model, initial design representations were based on geometric features, interpreted from CAD models using feature grammars. The enhanced framework allows the selection of design components from a pre-existing knowledge base. As before, stylistic change, defined by rule modifications (Knight, 1994) serves as the basis for rule replacement in the derivations of the original design with new rules that produce designs conforming to additional requirements.

For redesign, a domain-specific knowledge base is used to create a feature grammar based on the functional requirements of the design. This grammar is used to interpret an original design (a geometric CAD model) through a parsing process to create a design description of geometric features, if one does not already exist. Each rule of this grammar has an associated description that adds functional or behavioural information to that of the geometric features. The Function-Behaviour-Structure (FBS) model of design (Gero et al., 1992) is used to encapsulate this information (Figure 1).

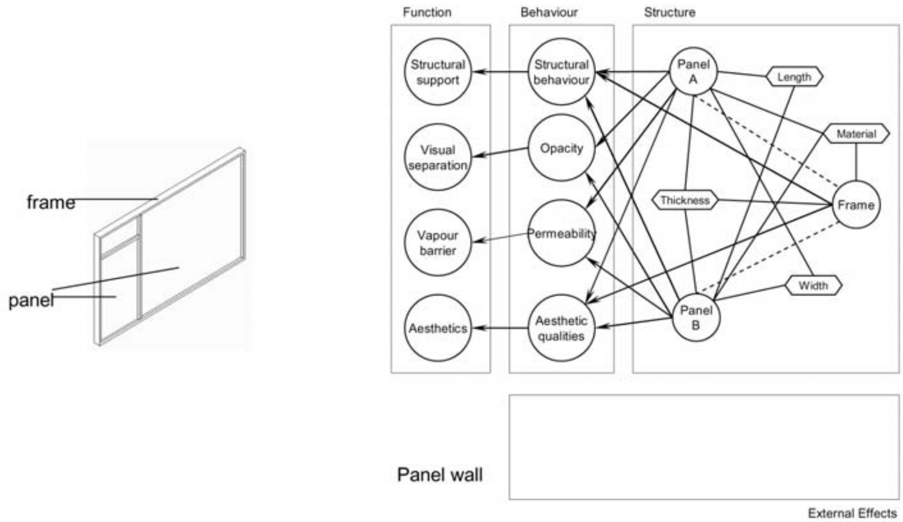

Figure 1. A panel wall and its corresponding FBS representation

The same knowledge base can be used to generate a set of rules that defines the underlying requirements for that particular design domain. Each of these rules has a similar functional or behavioural description associated with it.

The parse of the initial design based on the feature grammar is used to link the set of domain-specific rules (which typically have a different or additional set of requirements to the original design) to the original rule set. For every original rule used in the parse, its functional or behavioural description is indexed to the new rule set to provide alternatives to the existing design structures. An FBS network is used to define the search space within which the functional and behavioural indexes operate. By searching the indexed knowledge base of rules, then substituting rules in a derivation with those having the same function-behaviour descriptions, new feature structures (i.e. new designs) can be generated. This in essence represents a 'transformed' grammar which generates a new set of designs through a modification of the design state space. The basic steps in the process which are listed below can be seen in Figure 2.

- Original model created without an FBS description
- Feature recognition
- Adjacency subgraphs created
- Knowledge base searched for matching subgraphs
- FBS grammar rules extracted from knowledge base
- (Re)creation of original design with FBS descriptions
- Possible additional knowledge input from designer
- Redesign initiation: FBS nodes in design indexed for knowledge base search
- Search of knowledge base for alternative rules matching FBS descriptions
- Replace original FBS rules with alternative ones and generate new design

Implementation in GDL

GDL (Geometric Description Language) (Croser, 2001; http://www.gdltechnology.com:May 2002) is a scriptable language similar to BASIC. It is an open system that facilitates the construction of virtual

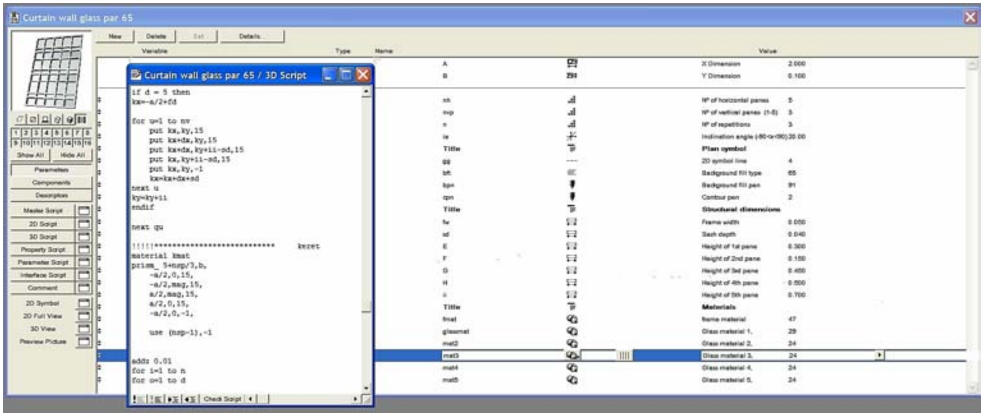

Figure 2. Framework for FBS description and redesign

objects which can be used in a variety of CAD environments. GDL objects can contain 2D and 3D geometric information, as well as non geometric attributes such as material, thermal, acoustic properties and cost (Figure 3).

GDL was developed by Graphisoft for use in its ArchiCAD™ software, where every model entity is a customisable GDL object. GDL objects are thereby used to represent building components such as doors, windows, walls, roofs, furniture, plumbing fixtures, HVAC systems, structural elements etc. These components can adapt to changing conditions through a customisable interface. GDL objects are currently used in two basic ways:

- As parametric building elements
- As manufacturer's components (currently a handful of manufacturers provide downloadable GDL objects from their product catalogs).

The implementation of our framework will use GDL objects as the CAD database components, with a library and knowledge base of GDL prototypes containing functional, behavioural and structural properties. An external inference engine will facilitate retrieval of applicable objects from the knowledge base, construction of grammar rules, and replacement of the associated objects in the existing design. Although we focus here on parametric building elements, this framework could also be used for design and redesign using manufacturer's components.

Because GDL objects can incorporate functional, behavioural and structural attributes, they can be introduced into the process as objects for inclusion in the knowledge base. This also implies that one could use the knowledge base to initiate design with FBS grammars using GDL objects from Step 5, bypassing the feature recognition stages.

Conclusions

Given that one change to a design can have a cascading effect, certain issues remain to be dealt with in such an implementation:

- constraint propagation: how does one manage local changes on a global scale?
- the interface to such a system: how does one control the redesign process? How is choice handled, when there is more than one possible object replacement? These issues are discussed in (Chase, 2002).

In summary, the redesign framework presented here represents a new application for grammar based methodologies, by combining them with FBS models. The incorporation of GDL objects into the framework as an existing and growing technology should facilitate its development, while expanding

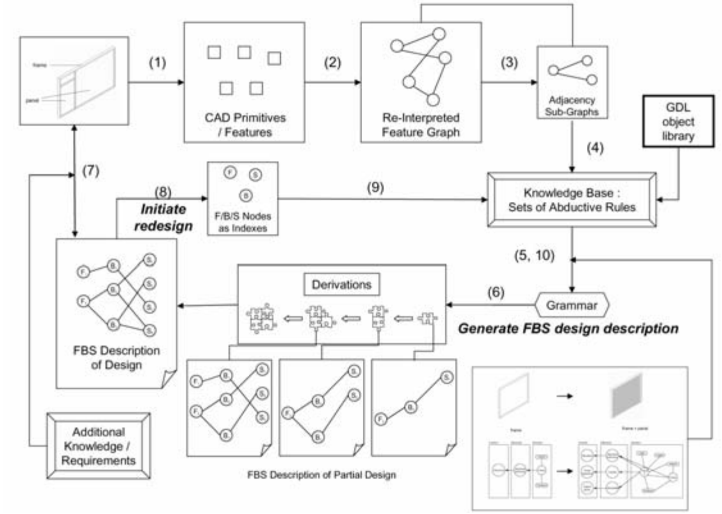

Figure 3. A curtain wall GDL library object in ArchiCAD

the potential uses of the paradigm by means of new knowledge bases.

References

Beesley P. and Seebohm T.: 2000, Digital Tectonic Design in D. Donath (ed), Promise and Reality: State of the Art versus State of Practice in Computing for the Design and Planning Process: 18th eCAADe Conference Proceedings, eCAADe & Bauhaus-Universität-Weimar, Weimar, pp. 287-290.

Chase, S. C.: 2002, A model for user interaction in grammar-based design systems, Automation in Construction, 11(2), pp. 161-172.

Chase S. C. and Liew P.: 2001, A Systematic Method for Redesign: Using function, behaviour and structure to facilitate grammar transformation, in H. Penttilä (ed), eCAADe19, Architectural Information Management: Proceedings of the 19th Conference on Education in Computer Aided Architectural Design in Europe, Otamedia Oy, Espoo, Finland, pp. 18-24.

Croser, J: 2001, GDL Object, The Architect's Journal, 14 June 2001, pp. 49-50.

Gero, J. S., Tham, K. W. and Lee, H. S.: 1992, Behaviour: A Link Between Function and Structure in Design, in D. C. Brown, M. Waldron and H. Yoshikawa (eds), Intelligent Computer Aided Design, North-Holland, Amsterdam, pp. 193-220.

Harfmann A. C.: 1993, Component-Based, Three-Dimensional "Working Drawings", in F. Morgan and R. W. Pohlman (eds), Education and Practice: The Critical Interface, Proceedings of ACADIA 1993, Association for Computer Aided Design in Architecture, pp. 141-151.

Knight, T. W.: 1994, Transformations in Design: A Formal Approach to Stylistic Change and Innovation in the Visual Arts, Cambridge University Press, Cambridge, England.

Liggett, R. S., Mitchell W. J. and Tan M.: 1992, Multi-Level Analysis and Optimization of Design, in Y. Kalay (ed), Evaluating and Predicting Design Performance, John Wiley, New York, pp. 251-269.

Mitchell, J. R. and Radford, A. D.: 1987, EAVE, a generative expert system for detailing, Environment and Planning B: Planning and Design, 14, pp. 281-292.

Starling, A. C. and Shea, K.: 2002, A Clock Grammar: The Use of a Parallel Grammar in Performance-Based Mechanical Synthesis, in Proceedings of DETC'02, American Society of Mechanical Engineers, Montreal, 10 pages.