Towards integrated performance-driven generative design tools

Article *in* Automation in Construction · March 2005

DOI: 10.1016/j.autcon.2004.07.002

Robert Aish
University College London

Towards Integrated Performance-Based Generative Design Tools

Kristina Shea1, Robert Aish2, Marina Gourtovaia1

1Engineering Design Centre, University of Cambridge, Cambridge,UK
http://www-edc.eng.cam.ac.uk/designsynthesis

2Bentley Systems, Exton, Pennsylvania, USA
http://www.bentley.com

Abstract. Generative design methods are capable of generating concepts and stimulating solutions based on robust and rigorous models of design conditions, design languages and design performance. The computer now becomes a design generator in addition to its more conventional role as draftsperson, visualizor, data checker and performance analyst. Motivated by the challenge to enable designers to easily develop meaningful input models of design intent to make best use of a structural generative method, this paper describes an initial combination of a generative design tool, eifForm, and an associative modeling system, Custom Objects, through the use of XML models. The current combined use is illustrated through an example involving generation of a set of 20 interrelated roof trusses with seven unique spans initiated by a parametric model of a saddle shaped stadium roof in Custom Objects. The paper concludes with a discussion of the synergies between associative modeling and generative systems and identifies future extensions aimed to exploit these synergies towards integrated performance-based generative design tools.

Keywords. generative design, parametric/associative geometry, advanced CAD tools, performance-based design, computational design

Introduction

The convergence of the geometric freedom offered by CAD tools and new manufacturing capabilities provide great opportunities for designers and architects to move away from traditional compositions based on symmetry and repetition in order to explore new forms. Digital design tools have now moved way beyond their early use, which was limited to the production of final drawings. An integrated design process involving CAD, solid modeling, a range of analysis tools, and rapid prototyping is emerging and has been essential in the design of today's most unique structures. With the recent development of parametric and associative geometry, CAD tools are now able to parametrically vary design concepts in step with designer intent. However, the fundamental geometric constructions remain static. The next phase in digital design lies in considering the computer as a collaborative partner in the design process capable of generating ideas and stimulating solutions in response to robust and rigorous models of design conditions and performance. Incorporating performance models to guide the generation process will yield tools that help architects, designers and engineers think critically yet qualitatively from different viewpoints, e.g. engineering performance, spatial performance, cost, and fabrication, throughout design conception of novel forms. However, providing tools that enable designers to easily develop these robust and rigorous models of design conditions and performance that guide design generation remains a challenge.

This paper gives an overview of combining a generative structural design tool, eifForm, with an associative modeling system, Custom Objects (CO). The initial integration described gives an opportunity to explore the complexity involved in combining such tools, the synergies between associative geometry and generative design as well as their combined potential for enhancing negotiation between architects and engineers in the development of novel yet efficient forms. A new way of designing architectural form and associated structure in parallel is emerging.

Parametric modeling as a concept and mathematical construct, e.g. parametric curves and surfaces, has been around for years with the first parametric CAD tools emerging in 1989. Parametric modeling involves the use of geometric constraints as well as dimensional relations and data to drive shape definition (Szalapaj, 2001). Values within parametric expressions can be modified by designers and are then propagated through the design, i.e. strategic manipulation. Alternatively, associative geometry allows dynamic manipulation of user-defined dependency relations by graphical manipulation, i.e. intuitive manipulation.

Often building on parametric concepts, generative design transforms the computer from a modeling assistant to a generator through the use of a described set of production "rules". Generative systems are aimed at both sparking new design ideas and solving difficult problems both of which provide design assistance and again extend designers' capabilities. In architectural design, shape grammars have been utilized for generating new spatial designs and characterizing designer style (Knight and Stiny, 2001), such as Siza's housing types (Duarte, 2003). Shape grammars (Stiny, 1980) can be basic or parametric where rules are described such that they are applicable to say any square or 4-sided polygons of any dimension, e.g. squares, rectangles, rhombi, depending on the allowed transformations. However, few generative systems in architecture encode parametric grammars (Heisserman, 1994) and in addition include performance feedback. One example can be found in (Caldas 2002) who uses a genetic algorithm combined with lighting and thermal analysis to generate novel and performance driven building envelopes and room configurations.

Custom Objects and eifForm

Custom Objects is a graph-based associative geometry modeling system that combines geometric modeling and programming (Aish, 2003). The project aims to provide a set of common parametric tools within Microstation and Triforma, the current CAD tools offered by Bentley Systems. While it includes capabilities common to other parametric modeling tools it also provides for intuitive, graphical manipulation of models and for reprogramming the tool. The latter is done through the creation of "custom objects feature classes" and is aimed at making more customized and responsive models.

The generative method within eifForm, called structural shape annealing, combines grammatical parametric shape generation, performance evaluation, which includes structural analysis, and stochastic optimization (Kirkpatrick et al. 1983) to support optimally directed exploration of discrete structural forms in relation to behavior, spatial and cost performance for both routine and challenging scenarios. Compared to published results in lightweight structures, the method is capable of generating efficient and innovative planar trusses (Shea and Cagan, 1999), single-layer space trusses (Shea and Cagan, 1997) and transmission towers (Shea and Smith, 2003). A recent extension to generating 3D trusses has resulted in a method that can balance spatial innovation with symmetry by using a planar truss grammar in combination with geometric transformations of planar designs and bracing algorithms.

Using eifForm effectively requires designers to create initial geometric, generative and performance models that reflects design intent. The generative model includes description of generative parameters, e.g. how many structural members to generate. The performance model includes structural considerations such as material type and loading as well as a mathematical description of design objectives. While the resulting designs produced by the process have gained great interest, a main challenge in integrating such a generative tool within a design process remains enabling designers to develop an understanding of how to create these models to best reflect the style of design desired, including encoding the necessary design constraints.

The motivation for integrating eifForm and Custom Objects is to enable:

• the development of richer design context models (currently geometry alone) in Custom Objects for transfer to eifForm as starting points for design generation,

• exploration of shape transformations in an associative geometry environment before allowing eifForm to automatically consider the many, often combinatorially explosive, design possibilities based on these transformations, and,

• effective visualization and manipulation of the geometric and topologic relations that are fundamental to the structural shape annealing method within eifForm.

Integration via eXtensible Markup Language (XML) models

A 'federated' systems architecture is used to relate the two tools, with eXtensible Markup Language (XML) employed for their integration. XML has become a language of choice for enabling integration of proprietary applications (Caporlette, 2000). XML provides flexible and adaptable information identification since it is a meta-language, i.e. a language for describing other languages, which allows one to design their own customized markup languages for most any type of document. Flexibility and interoperability make XML a favorite data format for two-dimensional web-graphics, archiving, encoding geotechnical information, 3D modeling and VRML rendering, as well as a growing interest in the AEC community. Other factors contributing to rapid growth in use of XML is that it is an open standard, has a good specification, is human-readable, is intuitive and flexible, can carry almost any type of information and off-the-shelf parsers are widely available for most common programming languages today, e.g. C, C++, C#, Java, Perl and Python.

In this integration, XML models of design scenarios, e.g. the description of a set of 20 roof trusses, are created in Custom Objects and then imported into eifForm to create the starting point for design generation. After the generative design process, eifForm then exports new XML models back directly into the original context within Custom Objects. The exchange of information between tools requires more than just transfer of geometric data since design intent within eifForm is formulated in terms of both geometric and topologic models of the initial design. XML has proven to be very beneficial for this purpose for the reasons given previously, even though there is currently no AEC standard for XML structural models.

XML does not provide a model for data, rather, it reflects the application's own data model. In the absence of a standard, each tool comes with its own XML model for essentially the same data. However, provided that tools document their XML formats well enough, writing converters between different XML formats is straightforward. This is the current level of integration here. One can envisage that a tool that converts automatically between different arbitrary XML documents with minimal pre-configuration by the user might appear soon.

The XML model file for eifForm (Fig. 1) consists, as it should be for any XML document, from one top-level document element, called EifForm, which encapsulates a number of nested elements. As it has been mentioned earlier, eifForm's input data has a hybrid character, and this is reflected in the four types of children elements of the EifForm element, namely, Settings, StructuralProperties, Obstacles and Design.

The Settings element lists the values and the names of settings that the user can use to influence the design generation process, i.e. the generative model. A manual for the program gives a full listing of possible settings and guidance on how to steer the design computation process in the desired and required direction by a thoughtful choice of settings and their values. Settings are (name, value) pairs, where name is the name of one of the global variables (definitions) in the program and value is the value that is assigned to the variable. Most settings are the attributes of the Settings element and are listed as a name="value" pair. A limited number of settings have a more complex nature and are represented by child elements within the Settings element, e.g. the buckling-constraint setting.

The StructuralProperties element defines the structural model and lists the material properties and a range of applied loads. The Obstacles element defines 3D geometric obstacles. The Design element describes the geometry, i.e. Points and Lines including references to the StructuralProperties element, as well as the topology of a design, i.e. Shapes. A Design also contains description of a local coordinate system defined in Custom Objects, CS. Together the four elements (Settings, Obstacles, StructuralProperties, Design) establish a relation between the geometry, topology, structural model and design generation settings.

The XML model in Custom Objects contains similar information as eifForm but in a different format (Figure 2). Here, XML serves as a script file executed or "played" within Custom Objects, each action described as a Transaction, rather than the more conventional use in eifForm as an input file. Topologic description of a design is encoded through the use of the Shape command. In Custom Objects the commands are executed sequentially producing an interlinked symbolic

Figure 1. An abridged eifForm XML file describing the starting point for a single stadium roof truss.

model and 3D graphical view (Figure 3). Once in Custom Objects the model can be reviewed and modified and then transferred back to eifForm.

Figure 2. An abridged Custom Objects XML model that describes the same design as in Figure 1.

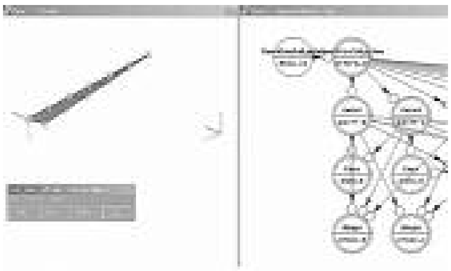


Figure 3. Sample initial single truss design within the stadium roof context and corresponding Custom Objects symbolic graph illustrating the key eifForm attributes, namely points, lines (in 3D cones) and shapes (shaded left although there is no structure there). Each design within the roof system is attached to it's own local coordinate system.

An example: a stadium roof truss system

To demonstrate the combination described and future potential, eifForm is used to generate the structural system, 20 interrelated trusses with seven unique spans, for a saddle-shaped stadium roof in relation to the design context defined by a parametric model in Custom Objects (CO). Each initial truss structure includes a starting shape defined by its location in the stadium, which is described by a reference to a unique local coordinate system in CO, span from 56.68 m to 72.89 m and support points along the rim of the stadium, constant for all spans at 6.07 m (Figure 4). The starting point can be either just a single triangle between the support points and truss tip or an initial lacing produced in CO (Figure 3). The trusses are generated using steel (S355) circular hollow sections (CHS) and respond to a loading across the top chord of the structure that estimates the distributed load from roof material and snow load as well as variable self-weight. Given an initial starting point, a new design is generated by altering a combination of variables including the original structural member sizes, either allowing continuous sizing or selecting from a predefined set of CHS sizes, node locations that in turn modify the truss shape, and structural configuration. The selected type of changes are made to the structure within an optimization loop that seeks to find optimally directed designs that adhere to modeled constraints, e.g. maximum stress, bucking and displacement (Figure 5). After generation the new designs are returned to Custom Objects so that they can be interpreted within the original design context.

In the stadium example, considering the longest span of 72.89 m, incrementally increasing the types of possible modifications in terms of structural sizes, shapes and topologies the structural mass is reduced. An initial optimization of member size alone within a fixed set of available sections yielded a mass of 48,283 kg while enabling a combined section size, shape and topology optimization led to a lower mass of 35,036 kg. In all cases geometric obstacles were used to fix the boundary of the structure so that it does not obstruct visibility in the stadium. The generative process can also be used to explore relaxation of constraints to understand their impact on the design. For example, changing only the constraint on discrete member sizes and using continuous sizing of CHS sections with a fixed ratio of tube outer diameter to thickness equal to 14 resulted in a set of new optimally directed topologies for the seven unique spans in the stadium context with structural mass ranging from 17,816 kg for the shortest span (56.68 m) to 41,003 kg for the longest span (72.89 m); see Figures 4 and 5. In this example an advantage of optimizing designs over a fixed set of section sizes was seen, even though it is a more difficult optimization problem, versus the pre-defined tube ratio chosen. While reducing structural mass may not be a primary design goal, it is used here as a

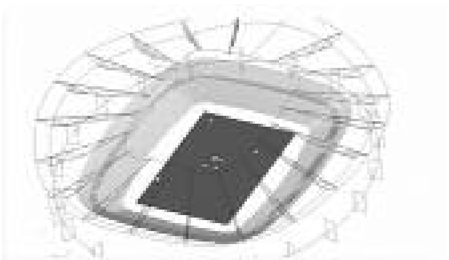


Figure 4. Set of 20 trusses with seven unique spans generated within the stadium roof context.

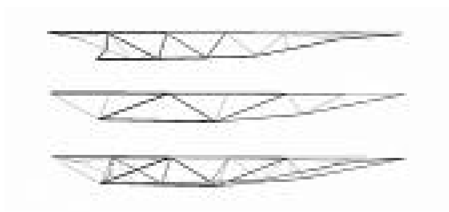


Figure 5. Two generated designs overlaid to illustrate the topology variation produced by changing the possible structural envelope and changing from continuous to discrete sizing. Relaxing both constraints led to a lighter, more regular structure, although less novel.

quantitative metric to compare design alternatives that all meet the required design criteria.

Discussion

A preliminary integration of two computational design tools, one generative and one associative geometry, through the use of XML models has been illustrated. Future extensions include:

- richer transfer of geometric data, parametric relations and constraints, as well as structural information between CO and eifForm,
- building within eifForm on the parametric geometric library offered in CO,
- improved integration of XML formats and parsers between the tools, and
- use of CO to gain better understanding of geometric transformation options when generating 3D trusses.

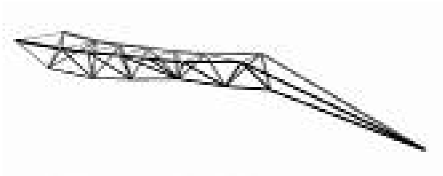
The exploration of stadium truss design alternatives was enhanced by being able to use the generative process within a larger parametric model of design context. It was found that due to the numerous constraints, most new designs produced did not vary drastically from the initial designs. In order to generate new designs that did not violate the design criteria of visibility, two geometric obstacles were placed below the truss producing a tapered solution. This does not occur naturally in the generation process as a deeper truss is most always lighter (Figure 5). Essentially the obstacles define the design envelope in rough way. A more robust and general way to implement design envelopes is underway and will allow joints to be moved and added relative to a predefined boundary. If the boundary changes then the related joint positions are updated. This technique has worked successfully in a specific implementation of structural shape annealing for transmission tower re-design (Shea and Smith, 2003). The numerous ways that points and curves can be related, added, and deleted in CO can then be used as a means to create the starting point for generation, gain understanding of possible transformations so that the generation process reflects desired style and transfer this generative model to eifForm.

Associative geometry provides the capability for simple manipulation of the parametric model to dynamically consider alternative 'what-if' sce-

Figure 6. A 3D cantilever design generated with eifForm.

narios. Through improved and richer integration of XML models between the tools, models can be created dynamically to take advantage of the parametric capabilities. For example, major geometry that defines the stadium roof can be altered, the new structural specification, including updating of loading models, sent to eifForm that then generates a new set of roof trusses in response to the new geometry. The integration then provides feedback within the parametric model on structural impacts to changing geometry.

Through this investigation and further conversations with structural designers it was discovered that most designers want a synthesis tool that generates designs which balance regularity and symmetry with spatial innovation. To achieve this effect without defining numerous types of symmetry constraints to maintain, a hierarchical composition model is used. Rather than generating 3D trusses using tetrahedrons, it has been found that it is more effective to generate a planar design (truss A), copy and geometrically transform it to produce the other half of a 3D truss (truss B) and use a bracing algorithm to tie the two halves together (Figure 6). Using the same planar grammar used to create the designs shown in Figures 4 and 5 and enabling several possible geometric transformations of the design copy, e.g. translation, rotation, reflection, scale and their compositions, as well as using several bracing algorithms, a fairly comprehensive language of 3D truss structures can be produced from a very simple structural grammar. An example, suitable for the stadium roof truss scenario, is shown (Figure 6) that was generated by reflecting the copied truss, B, about a skewed vertical plane defined by the lower truss chord. Again, understanding of the geometric dependencies and transformations used in the generative method will be key to it's effective use. Thus, further integration with Custom Objects to take advantage of the geo-



metric libraries available will be pursued.

Conclusion

Integrated performance-driven design tools are aimed at creating new design processes and exploiting computing capabilities for stimulating novel yet achievable solutions. Integrating a generative design tool within a CAD environment will allow designers to experiment with generative methods within a familiar context and also give more freedom to create complicated geometric forms that describe the spatial aspects of the generative design context. Adding the extra capabilities of associative geometry and generative structural design will enable designers to experiment with many different design scenarios and dynamically assess the structural impact of alternative global forms. The real challenge is to make tools that designers want to use in order to explore the potential for performance-based tools to aid negotiations in multi-disciplinary design teams.

Acknowledgements

The first author would like to thank the EPSRC (UK), a Philip Leverhulme Prize awarded through the Leverhulme Trust (UK) and Bentley Systems for funding this research.

References

Aish, R.: 2003, "Extensible Computational Design Tools for Exploratory Architecture", in B. Kolarevic, (ed), Architecture in the Digital Age, Spon Press, forthcoming.

Caldas, L: 2002, “Evolving Three-Dimensional Architecture Form”, Artificial Intelligence in Design '02, Cambridge, United Kingdom, pp. 351-370.

Caporlette, B.: 2000, “Application Integration Using XML”, Proceedings of XML Europe 2000 (http://www.gca.org/papers/xmleurope2000/index.html), Paris, France, 12-16 June.

Duarte, JP: 2003, “Customizing Mass Housing: The grammar of Alvaro Siza's Houses at Malagueira”, Environment and Planning B, forthcoming.

Heisserman J: 1994, "Generative Geometric Design", IEEE Computer Graphics and Applications, 14(2):37-45.

Knight, T. and G. Stiny: 2001, “Classical and non-classical computation”, Architectural Research Quarterly, 5(4):355-372.

Shea, K., and J. Cagan: 1999, “The Design of Novel Roof Trusses with Shape Annealing: Assessing the Ability of a Computational Method in Aiding Structural Designers with Varying Design Intent”, Design Studies, 20:3-23.

Shea, K and Cagan, J: 1997, “Innovative dome design: Applying geodesic patterns with shape annealing”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 11, pp. 379-394.

Shea, K and I. Smith: 2003, “Improving Full-Scale Transmission Tower Design Through Topology And Shape Optimization”, submitted to the ASCE Journal of Structural Engineering.

Stiny, G. (1980), "Introduction to Shape and Shape Grammars", Environment and Planning B, 7:343-351.

Swartz W., and C. Sechen: 1990, “New Algorithms for the Placement and Routing of Macro Cells,” in Proceedings of the IEEE Conference on Computer-Aided Design, Santa Clara, CA, November 11-15, IEEE proceedings: Cat No. 90CH2924-9, pp. 336-339.

Szalapaj, P.: 2001, CAD Principles for Architectural Design, Architectural Press, Oxford.