Utilizing customizable generative design tools in digital design studio: XpGEN experimental form generator

Burak Pak, Ozan Onder Ozener and Arzu Erdem
In this paper, we present a generative design model for conceptual design in architecture. Based on this model we developed and implemented a compact, open-ended generative tool with a connected design evaluation database. Core concept of our generative approach is to achieve complex forms from a base primitive and create the form from the modified instances.

Our tool is used in various levels of design studios, including graduate and undergraduate students. Designs from these experiments are evaluated in a qualitative framework.
I. INTRODUCTION

Digitalization of the design tools and design process has currently shifted the conceptual thinking both in real (physical) and virtual architecture. It is clear that the new paradigm of emerging design process is bound to electronic and digital technologies. The other fact in digitalized architecture is the base with an extremely fluid and (n) dimensioned application space. Thus, we need new and capable tools to practice in this new environment.

Besides supporting the designer and forming the starting points for conceptual design in digital environment, generative design is a new approach in digital and virtual architecture. Computer supported generative systems can provide useful inputs for developing different types of architectural products. Usage of generative systems in architectural design is an expanding domain. This research and design activity includes parallel processes like algorithm creation, software development and form design. Connection of the algorithm and the form is a challenge for the designer who must predict the topology and the geometry of the resulting forms while designing an effective generative algorithm.

1.1. Previous work

Since the pioneering research in Palladian grammar by Stiny and Mitchell [1], Queen Anne Houses grammar by Flemming [2] and computational form analysis by Robert and Rivka Oxman [3], a variety of commercial modeling and animation software have become widespread. Computers have been transformed from supporting tools to design environments.

Convergence of computer aided technological processes and biological models of growth led to the development and transformation analysis using animation rather than conventional architectural design software. [4]

Greg Lynn’s animated forms are one of the first form generation experiments to explore the linkages between characteristics of time, topology, and parameters that combine to establish the virtual possibilities for designing in an animate rather than static space. Lynn’s dynamic modeling systems are based on the interaction of multiple parameters calculated sequentially rather than in an instant. [5]

“Our World” Project is a contemporary example of experiments that seek possibilities of using animated parametric modeling with architectural ideation. The project empowered subsequent cataloguing of animated topological hybrids that allowed designers to evaluate multiple forms. “Our World” Project is important because it introduces the power of computer-user interaction and evaluation. [6]

Paramorph-I is another parametric modeling software developed by Mark Burry and Grant Dunlop in which associated geometric strategies are used as a speculative design tool. The accidental effects achieved during the process provoked users to create new designs. [7]
“MorphZshapes”, developed by Kostas Terzidis, is a similar software that morphs topologies to generate hybrid forms. Through the system, user can modify and control the flow of compositional evolution and replay it many times by varying all of the transformational parameters. [8]

Experimental works of certain research groups and professional initiatives clearly demonstrates the potential of design-driven algorithms and generative design methods in architectural design but results from these experimental works do not provide sound methods to be used for conventional practice yet. Algorithmic form generation in architectural design is still one of the significant research areas in design computing.

2. MOTIVATION

Experimenting with new tools and techniques is vital for education: it is experimental architects’ use of innovative techniques to generate new organizational processes that enables them to understand the possibilities contained with the design process. As a consequence, the deterministic notions of causality are replaced with nonlinear, bottom-up systemic processes that produce emergent effects. [9] It is essential to develop theoretical frameworks and computational environments to relate computational thinking to design process. [10]

Our motivation is coming from basic design principles to have a complex structure with multiplying a base primitive object. Although this approach is widely used for creating structures with linear shape grammar methods, we approached the generation process as a non-linear random process. This approach implies a significant distinction from the shape grammar based creation methods. In shape grammars framework computational problems related to encoding rules and their execution in a computer program took precedence over user interfaces. Thus, these systems were not suitable, or ready, for general use by non-programmers, novice users of shape grammars, or design practitioners. [11] Most work on shape grammar implementations has involved straight lines or planar surfaces. [12]

In contrast, generative approach allows designer to have partially unexpected design results with an acceptable level of prediction. This nature of the generative approach gives user more flexibility and variety over forms.

Our main research problem is to create a compact generative tool that meets the needs of conceptual shape generation in the design studio. We have set some criteria for the tool implementation and aims for the usage.

Criteria set for the tool:
• Support different types of surface structures like polygonal, NURBS or subdivision surfaces.
• Build a modular code structure for further development and modification.
• Provide significant control on random based shape generation via user input and settings.
• Allow any primitive as a starting object
• Utilize database for evaluation

In order to increase the variety of the output, allow easy evaluation and to provide significant control, we have employed several operations to generate forms. These are:

• Transformation (Translation, rotation and scaling using new parameters)
• Multiplication (Clone operation)
• Reformation (Translation, rotation and scaling using the existing parameters)
• Modification (Topological changes)
• Animation (Evaluation during the transition process)

Within this research framework it is crucial to demonstrate the potentials of the generative tools for creating architectural design input. Time dimension is also important to illustrate the creation process and to enhance the design with animations and still images.

Objectives on the usage side include the observation of the designer reaction and design thinking within generative framework as well as getting feedback for further improvement of the proposed tool.

3. METHODOLOGY AND THE MODEL

In this model, the starting object and transformation rules are not restricted and can be determined by the user. Evaluation of the generated products is be made by the user(s) with connected modules of Xp-GEN. The model consists of three main modules,

1. Generator
2. Design file manager
3. Xp-DAT database

Generator module executes the user defined rules to the starting object. In this version modification functions are controlled by the production engine, the engine allows to use modifiers with different combinations. With these modifiers and user defined parameters the user can try different types of transformations and randomized modifications.

The model supports both linear and randomized generation strategy.

In random generation process, first phase is to define object and space relationship with the limits of x, y and z coordinates. These settings are defined by maximum limits of desired design space. In the modification menu there are 4 movable modifications; Scale, Twist, Bend and Skew. The other parameters can be listed as;

• Coordinate parameters
• Rotation parameters
• Number of copies
• Name of the starting object

These known and simple modifications are selected to maximize the
eligibility and usability of the tool in the design development process. These are the basic object modification rules that can be found in most digital design tools. In the random creation phase, starting objects are transformed using these modifiers using random variables generated in the script. Also objects are multiplied and positioned within the borders of design space or a defined route. It is also aimed to produce non-platonic shapes for developing designs. The flow diagram is given in Figure 1.

**Figure 1: Xp-Gen Flow Diagram.**

Xp-GEN also supports a linear shape creation process. All the parameters defined above can be pre-determined by the user. This feature allows a tightly controlled generation process. In generative design systems, designers can be overwhelmed by the amount of information they are able to generate in a short time. This process is as though designers are exploring a forest in the dark with a flashlight. [13] File management and storage modules in Xp-GEN are simple solutions to overcome this problem.

Generated objects can be stored and loaded using the design file manager module. This module is a standalone script that generates a thumbnail of the alternative, scans the directory, finds all alternative file names with the same base name that already exists, takes their count, adds 1 to it and saves a new file and thumbnail with an incremental number. Xp-DAT database is formed using a simple script that collects all files in the directories and classifies them. Xp-DAT provides a reporting feature that allows users to compare and evaluate different design alternatives altogether. Another function of the module is to manage huge number of alternatives and media generated by the user.
4. IMPLEMENTATION

The tool is implemented in 3D Studio MAX with MAXScript Language and MAX SDK. Additional functions or modifiers can be added to plug-in based code structure.

Xp-GEN generation module starts with definition of object primitives given in the scene. These primitives are defined by set pointers. Other functions also have assigned variables like number copies, coordinates and affine transformations. Simple user interface includes, base object selection, settings and parameters, creation and editing parts (Figure 2).

- Parameter part takes the values for space limits number of copies and rotation amount.
- Modification part consists of built in modifier selection menu. User can select the one or multiple modifiers to be assigned on the base objects and the instances. The key point in this menu is the randomized or linear creation selection.
- Editing part is crucial to evaluate the resulting form and alternatives. User can easily tweak settings, remove any modifier, or copies. After changing the settings user can update the model according to new input.

Database connection is created with a macro plug-in. the main function of this module is to store the design alternative and recall it from the design database. We used a simple relational database structure that stores the
created design alternative with the following table structure.

- Design ID number
- Thumbnail image created in generation environment
- Base primitive object file link
- Number copies
- Space limits
- Rotation and transformation values
- Boolean flags for each modifier
- Final design model file link

This simple structure in MS Access gives user the essential functions of the database like sorting, filtering and reporting.

4.1. Design studio structure and workshops

In order to explicitly discuss the cons and pros of Xp-GEN at work, we have implemented three workshops at Istanbul Technical University, Faculty of Architecture in graduate and undergraduate levels. Our aim was to test our prototype tool in real studio context and search for empirical evidences for its potential to be used for creative purposes in the design studio. We undertook a qualitative study using systematical observation methods (Sommer & Sommer, 2002). We searched evidences for student reaction to prototype tool in terms of usability and learning, information about the design process, also quantity and complexity of starting objects and final design alternatives.

All of these experiments were short term activities with 8 to 15 students for each experiment. Regardless of the studio level, we held two tutorial sessions for the basics of generative systems and usage of our tool. We gave a simple design problem in each experiment. For data gathering and evaluation, every design alternative is stored in our Xp-DAT database module and reviewed after each studio session (Figure 3).

Workshop 1 was an urban experiment in Istanbul. Students were given a limited space in an urban configuration. The key problem was to design own primitive object and show variations using the plug-in. Main evaluation criteria was resulting form must demonstrate a level of relation to the existing texture of the given environment as well as the contrast inherited from the design process.
Workshop 2 was an abstract living space for cyber inhabitants. Students were encouraged to create an abstract living space with their own scenario about an imaginary inhabitant. Regarding to the created scenario, form and function problem was also determined by the designer.

Workshop 3, da351, was integrated into first year basic design studio in 2003. In contrast with the previous workshops, a predefined design problem wasn’t introduced to the students. The medium and the tool was the problem itself. They were free to create their own concepts to generate animated parametric forms with Xp-GEN. Novice users’ interaction patterns and design development strategies were explored.
4.2. Data Analysis and Evaluation

We undertook a qualitative study using systematical observation methods. Additionally, we have analyzed and sorted the data gathered in Xp-DAT database. We conclude that students were able to use the tool after the first tutorial session and they gained proficiency in the second studio gathering. They improved the designs while producing more alternatives (Table 1).

Except two students in workshop 3, all of the students were familiar with the 3D Studio MAX software environment. We observed that all of the students were able to distinguish the plug-in interface and usage procedures from the commercial software for the given particular problem. From the written feedbacks, they reported design process with our tool have two distinct components: designing the starting object and using the raw object for generative process. It is also reported that students used other functions of the commercial software for design development and they benefited from the plug-in structure of our tool for further modification, visualization, animation without data conversion to external software.

<table>
<thead>
<tr>
<th>Workshop</th>
<th>Number of Students</th>
<th>Tutorial Session 1 Mean of Design Alternatives</th>
<th>Tutorial Session 2 Mean of Final Design Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workshop 1</td>
<td>14</td>
<td>0.928</td>
<td>1.785</td>
</tr>
<tr>
<td>Workshop 2</td>
<td>15</td>
<td>0.866</td>
<td>2.06</td>
</tr>
<tr>
<td>Workshop 3</td>
<td>8</td>
<td>2.5</td>
<td>3.375</td>
</tr>
</tbody>
</table>

Regardless of studio problem and level, all students understood the final design is bound to starting object and they designed several alternatives for the initial step to achieve different design alternatives. Complexity of these
starting objects was varying from very simple to moderate. Students developed their own conventions like avoiding complex starting objects with huge number of faces and limiting the number of clones.

All of the students used the animation tool to illustrate the process as well as evaluating the each time span as a still design alternative. Observation data from Table 2 shows that they utilized the tool to create a large number of design alternatives and focused on limiting the alternatives for final design decision. In order to evaluate the high number of alternatives, they used our visual design database. This shows that students were able to understand the concepts and power of generative processes as well as using the given generative method to achieve unique design output.

<table>
<thead>
<tr>
<th>Workshop</th>
<th>Number of Students</th>
<th>Mean of Starting Objects</th>
<th>Mean of Design Alternatives</th>
<th>Mean of Final Design Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workshop 1</td>
<td>14</td>
<td>5.21</td>
<td>23.285</td>
<td>2.214</td>
</tr>
<tr>
<td>Workshop 2</td>
<td>15</td>
<td>7.4</td>
<td>19.333</td>
<td>2.2</td>
</tr>
<tr>
<td>Workshop 3</td>
<td>8</td>
<td>3.25</td>
<td>30.75</td>
<td>3.125</td>
</tr>
</tbody>
</table>

5. CONCLUSION AND FUTURE WORK

The research showed that usage of simple generative interfaces for educational purpose returns acceptable results. In these experiments, it has been observed that the tool supported the object and the structure creation process in digital environment also gave an opportunity to try and evaluate a variety of design alternatives. It is also observed that tool has increased the students’ capacity to visualize abstract 3D symbolic structures as well as modification of digital objects.

Consequently, the complexity of digital 3D structures became a clear entity and easy to understood. Animating the generation and including “time” as a parameter of the design process made the process more attractive and more effective. The plug-in has motivated them to develop interesting concepts, expressing themselves and combining different types of media.

The design file manager module improved and expanded 3d Studio’s default file loader with the poor quality thumbnails. Students were able sort the designs according to their creation order and their ID’s. The naming convention and high quality thumbnails made it easy for the students to cope with enormous number of outcomes. Additionally, Xp-DAT database and its reporting capabilities linked with the design file manager allowed students to evaluate generated objects and compare them visually.

We also came up with certain improvement feedback from the studio experiment. Being a simple and clear plug-in, our interface does not avoid self-intersections of the 3D surfaces or problematic creations. The future
work may include a controller and support module to give feedback to the user about the topology and geometry of the resulting forms. It is also possible to implement different set operations to achieve design variety and alternatives.

While using Xp-GEN, students were able to develop new design concepts. On the other hand, it was hard for the students to superpose functional program with the parametric forms. In some cases, students recomposed the generated objects to overcome this problem.

The set of operators developed were sufficient for the students. Multiplication, transformation, animation and reformation were the most common operations used in the design phase. We have to note that irregular forms generated by modification operations worked better in basic design studio rather than the advanced studios.

References
Utilizing customizable generative design tools in digital design studio

Burak Pak
Istanbul Technical University
Institute of Informatics, IT in Design Graduate Program
ITU Taskisla Campus, 34437, Taksim / Istanbul, Turkey
burak.pak@itu.edu.tr

Ozan Onder Ozener
Texas A&M University,
Department of Architecture
TX 77843-3137, College Station, USA
ozener@tamu.edu

Arzu Erdem
Istanbul Technical University
Department of Architecture
ITU Taskisla Campus, 34437, Taksim / Istanbul, Turkey
erdemar@itu.edu.tr