AGENCY GP: Genetic Programming for Architectural Design

Peter Testa, School of Architecture and Planning, Massachusetts Institute of Technology, USA
Una-May O’Reilly, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA
Simon Greenwold, Emergent Design Group, Massachusetts Institute of Technology, USA

Abstract
AGENCY GP is a prototype for a system using genetic programming (GP) for architectural design exploration. Its software structure is noteworthy for its integration into a high-end three-dimensional modeling environment, its allowance for direct user interruption of evolution and reintegration of phenotypically modified individuals, and its agent-based evaluation of fitness.

1 Overview
AGENCY GP is a software framework we are developing to explore the possibilities for architectural design of offices and workspaces that arise from new concepts for organization and management theory that include non-hierarchical and emergent organizations.

1.1 Maya Integration
Through the Alias|Wavefront Maya platform’s API, we are building a genetic programming system operating over a language capable of expressing three dimensional designs and the free-form deformation of space to create morphologies that a designer may not have otherwise imagined possible. The Maya platform allows us to abstract the representation of three-dimensional forms so that we can operate freely on them without concern for the complexity of the underlying geometry. The language we have developed manipulates spatial constructs at a high enough level that its individual operations are meaningful to a designer. This language, in its representational power combined with its simplicity of expression, is the first major innovation of AGENCY GP.

1.2 User Control
Typically interruption, intervention, and resumption (IIR) of the evolutionary process is difficult to achieve in genetic programming environments because in most systems it is impossible to map changes of the external (phenotypic) individual back onto the internal genotype. However, because of the high level of transparency of our GP language, we have been able to design a system that will allow for IIR. A designer will be able to employ statements of the language himself to manually alter the forms of members of the population and reintegrate them for continued evolution. IIR, the second software innovation of AGENCY GP is a primary objective of research, and a major area for our continued investigation.

1.3 Agents
The third innovation AGENCY GP will employ is the determination of fitness from the point of view of various agents that inhabit the space. Agents are not necessarily single users; they may also represent emergent organizational elements such as a group of users who express a coherent need,
a resource that has allocation demands, or a group of resources that provide a service. An agent may represent the pattern of a group, its needs for privacy, meeting space and collaborative surfaces, or it may undertake the concern of management structure or productivity. What agent-based fitness allows for is a modular structure for the integration of multiple criteria for fitness. We intend to abstract the agent structure so that new agents may be developed and employed for new applications without rewriting the entire system. This agent-based evaluation of fitness is well suited to expressing the conflicting, non-linear, multi-level spatial requirements of emergent organizational structures.

2 Background and Motivation
AGENCY GP is the software arm of a larger project called AGENCY. The project seeks architectural responses to the radical transformation that business organizations are currently undergoing. The pace of organizational change is being driven by the rapid development of commercial technology, global markets and reengineered, quality oriented organizations. This constant need to change gives rise to organizations that are no longer stable, but continuously adapting to their shifting environments. Such organizations can be said to be “emergent” and include many of today’s commercial and governmental organizations.

Vertically structured office buildings no longer provide the model for most business organizations. With the advent of widespread use of telecommunications, information technology, and corporate reorganization in the 1990’s, new forces are actively reshaping the architecture of office buildings. There is a shift in the United States toward research and development, management and finance, consultancy, and the culture industry, productive activities less prone to standardization and bureaucratization. Driven by the demand to improve office productivity, businesses and organizations have begun to experiment with a variety of alternative office methods. However, there exist no working models of an intelligent adaptive architecture. The AGENCY project focuses on application oriented basic research to develop new design software that generatively models the complex interaction of physical space and information technology within emergent organizations. Using the research software this project will also generatively design and test spatial systems and work environments for emergent organizations.

We have chosen genetic programming to address these challenges because the strengths of the model are well matched to our system’s desired characteristics. First, we are aware of the impossibility of modeling emergent organizations deterministically. To try to design a top-down algorithm for the creation of workspaces would certainly fall victim to our inability to name every constraint the problem entails. Therefore we look to the bottom-up solutions, Artificial Life (ALife), and in particular, GP affords, to construct solutions that are consistently sensitive to complicated interactions that a user need not explicitly codify. Second, we are interested in the genetic model’s ability to offer a user an entire population of solutions to peruse and potentially to reevaluate. We are aware the process we are involved in is not a simple optimization with a single goal, but has many potential fruitful avenues of exploration. The multi-tracked exploratory process of population evolution provides a designer multiple alternatives with which to interact at any point. Our goal is to develop GP as a design partner, offering options that would otherwise not come to light.

3 Methodology
3.1 Tool Design and Implementation
We are implementing our software as a plug-in extension of Alias|Wavefront’s Maya package, a leading tool for the creation and visualization of complex virtual environments. Maya is of widespread use in the fields of three-dimensional animation for film and broadcast, but has only recently begun to see broad application to explorations of morphology such as AGENCY GP. Maya’s open architecture allows for software developed using all modern high-level languages features to exploit the power of its inbuilt library of three-dimensional operations. Any imaginable three-dimensional form is expressible in terms of Maya’s available spatial transformations and Boolean operations.

3.2 System Overview
Our implementations of the GP’s main generational loop, selection, mutation, and crossover are conventional. The individuals in the population, described below, are fairly complicated structures. Some of the data they contain is of fixed length and type, and is therefore evolved in place, while some of it is in the form of a variable-length sequence of language operations. Therefore we use a hybrid GA/GP to manipulate both types of values.
3.3 Representation and Language

We use a combination of Maya and C++ objects for the internal representation of individuals in our population. A user begins the process with a Maya scene containing one or more closed NURBS (non-uniform rational b-spline) curves selected. These curves should be coplanar but may be of any closed form and may intersect. In the interpreted phenotypic representation, these curves will be extruded into space. (Fig. 1-2)

When the AGENCY GP is invoked from within Maya, a C++ object is permanently attached to each NURBS curve. This C++ construct contains evolvable values pertaining to the shape and architectural function of the region the curve encloses. The object also contains an evolvable sequence of operations in our GP language to be applied to this curve. Each curve is given a height of extrusion and treated as a NURBS surface extending from the base plane into space.

The operations in our language are simple but powerful transformations of these NURBS surfaces. The images below demonstrate the operations of our language applied to the left-hand NURBS surface from the scene. (Fig. 3-8)

These operations form the core commands of our language. Mutation may consist of addition or deletion of an operation, or the change of a parameter. Execution is strictly linear; there is no facility for conditionals or branching. This simple program structure contributes to our ability to implement IIR.

We are counting on these surfaces to intersect with each other in ways we cannot predict. Each Boolean operation we apply—intersection, union, or subtraction—forms a new enclosed surface that may be assigned its own architectural function based on values in the GA section of the curve.

Each individual in a population, therefore, is comprised of several separate Maya/C++ objects each of which carries a NURBS surface, certain evolvable values, and a list of operations. That we never directly query or modify the low-level geometry of the Maya objects, but allow Maya to perform all needed transformations and Boolean operations is what makes the language high enough level to be useful for the direct intervention of a designer.

3.4 Agent-Based Fitness

Integration of GP with a high-end three-dimensional modeling tool allows us to apply extremely abstract and computationally expensive heuristics as measures of fitness. We are developing a framework into which individually developed software modules can be placed that evaluate our designs from various perspectives. These modules are called agents because each has a specific agenda for determining fitness. Some agents may represent actual users of the space, while others will be interested in issues such as fire-code compliance, or energy efficiency. Virtually any criterion for evaluation can be coded and dropped in as an agent to our framework. We can specify that workspaces require a certain quotient of natural light or that circulation spaces desire width enough to allow for conversation. Using agent-based evaluation, we will be able to model management structures and determine their influence on potential designs.

The fitness of a design will be determined by combining the various satisfactions of the agents deployed into the space. We will be able to integrate any calculable metric to our system through this architecture, so that AGENCY GP’s richness and power can grow incrementally as we continue to develop agents.

3.5 Interruption, Intervention, and Resumption

Once a population has been ranked by fitness, the AGENCY GP becomes open to IIR. The entire population of interpreted designs is available for viewing by the user, who has several options. The user may simply re-rank individuals and allow evolution to continue, or he may take a candidate and apply one or more operations from our language to as many of the NURBS surfaces that com-
prise it as he wishes. The transformations the user applies will be added to
the list of operations in the internal representation of the individual. By
providing the basic operations of three-dimensional modeling through
our language we enable designers to make targeted modifications of de-
signs before allowing evolution to continue.

4 Conclusions

We have demonstrated that it is possible to outline a methodology for the
integration of artificial life into architectural design exploration that
empowers both the programmer and user of the system to influence the
process.

5 Future Software Research

Our first order of business is to continue the implementation of the proposed system. When com-
plete we will be able to devote our attention to the creation of a variety of agents suited to different
tasks. The issue of user-intervention opens up research possibilities into how subjective selection
interacts with computed fitness. Which agents, for instance, are the ones that a designer tends to
favor? Are there agents who are difficult to satisfy simultaneously? Observational and statistical
analysis of agent-based fitness and user selection will certainly lead to interesting insights into our
software and our preferences as designers.

References

Bonabeau, E.W. (1997). From Classical Models of Morphogenesis to Agent-Based Models of Pat-
tern Formation. In Artificial Life, vol. 3, 199-211.

ing, MA: Addison-Wesley Publishing Company Inc.


Making Sense of the World. In Journal of Science Education and Technology, vol. 8, no. 2,
3-19.
Figure 6. CUT (START, STOP).

Figure 7. Boolean Intersection.

Figure 8. Boolean Subtraction.