Exploring generative growth and evolutionary computation in architectural design

Martin Hemberg[1,2], Una-May O'Reilly[3], Achim Menges[2], Katrin Jonas[4], Michel da Costa Gonçalves[2], and Steve R Fuchs[5]

[1] Department of Bioengineering, Imperial College London
martin.hemberg@imperial.ac.uk
[2] Emergent Design and Technologies, Architectural Association
achimmenges@aaschool.ac.uk, mdcg@aaschool.ac.uk
[3] Computer Science and Artificial Intelligence Lab, MIT unamay@csail.mit.edu
[4] Adaptive Architecture and Computation, Bartlett School of Architecture, University College London katrin.jonas@ucl.ac.uk
[5] Southern California Institute of Architecture steve@virtual-architect.com

1 Introduction

In this chapter we investigate how ideas from Artificial Life and evolutionary computation can be applied to one facet in the domain of Architecture. The human eye is intuitively drawn to the organic shapes of sand dunes, curling vines, rolling hills and other natural phenomena. Because of its strong aesthetic appeal, our particular interest is in generating biologically inspired form for architects. In the past, architectural form was constrained by material and structure and was able to reflect only a small degree of natural form in examples such as rounded pillars and domed roofs. In an exciting paradigm shift in architecture, contemporary computer aided design and manufacturing in interaction with integrated, human designed materials have largely unleashed today's architects from these shackles. They can now move beyond simply appreciating the graceful form of an emerging flower that bends in response to the sun's position or admiring the evolved shape of a natural shelter that responds to seasonal elements.

Beyond the aesthetic appeal delivered by natural form, conveniently such form is often very efficient in terms of structural capacity and economy of materials. D'Arcy W. Thompson, [1], observed "The form of any particle of matter, whether it be living or dead, and the changes in form which are apparent in its movements and in its growth, may in all cases be described as due to the action of force". Unlike most man-made designs, structures found in nature are often robust to a wide range of failures and they can fulfill multiple functions.

Thompson famously counseled that form follows function. Likewise, for achieving form akin to nature, our approach has been to have form follow process – the growth and evolutionary process that occurs in the living world. Essentially, this approach of mimicry allows designers to capitalize on Nature's strategies which, arguably, are the most compelling means of achieving Nature's outcome.

The manifestation of this conceptual statement of our goals is a computational design tool named Genr8. Genr8 allows an architect or designer to both *grow* and *evolve* three dimensional *digital* forms or surfaces. At the core of Genr8 is a "growth engine" that executes and visualizes a set of growth instructions (more formally a "HEMLS" or Hemberg Extended Map L-system grammar). When Genr8 executes a set of growth instructions, Genr8 mimics growth by expanding a planar surface (specified by the instructions) much like a primitive cell of a leaf expands into a complete leaf. Just as a growing leaf twists and shapes itself in response to environmental factors such as gravity and sunlight, Genr8's expanding (digital) planar surface grows in reaction to an environment of digital physical boundaries, attractors and repellers. In concert with Genr8's growth engine, an evolutionary algorithm selects, genetically varies and adapts the growth instructions and their resulting surfaces. This exploitation of evolution relieves the architect of the cognitively cumbersome task of providing Genr8 with a specific set of growth instructions.

In an effort to capture the combined technical and application aspects of Genr8, this chapter is co-authored by Genr8's developers (Hemberg and O'Reilly) and architects who have used Genr8 (Fuchs, Gonçalves, Jonas and Menges). In Section 2 we set the context for Genr8 in terms of related work. Section 3 describes the organic growth algorithm at the core of Genr8 and its evolutionary algorithm. We aspire to make these comprehensible to a reader who may be either a developer or a technically adept designer. A graphical user interface and scripting language, described in Section 4, are the interface to design control. We then devote the remainder of the chapter to examining designers' experiences with Genr8. Each experience has multiple facets. In Section 5 the authors who are architects are given voice. From goal, through methodology and to outcome, each describes, with illustrations, how he or she conducted a project with Genr8. This informative material then sets the stage for a discussion in Section 6 where the developers and architects collectively explore their impressions of Genr8 effectiveness, the sources of that effectiveness and its general implications for architectural form design.

2 Related Work

A number of architectural design groups have explored generative or growth processes. One of the early pioneers was John Frazer who began his work at the Architectural Association (AA) in the 1960's [2]. Throughout the years Frazer has been involved in a large number of projects exploring generative concepts. Particularly noteworthy is his Universal Constructor which is based on Cellular Automata (CA) [3]. Interestingly, many of his experiments are direct physical implementations of growth algorithms or evolutionary algorithms using custom-built hardware, sensors and actuators. Another example is a predecessor to Genr8, the MOSS project by the Emergent Design Group (EDG) at MIT. MOSS explored the use of hand designed L-systems that were digitally visualized. The L-system grammars were quite restricted in their syntax to allow a planar region to be delineated by segments because L-systems rather than Map L-systems were employed. A project resembling Genr8 was undertaken by

Coates and co-workers [4]. They combine L-systems and Genetic Programming and grow 3D forms on an isospatial grid influenced by an environment. Another interesting result is the work by Hornby and Pollack which demonstrates the advantages of using generative encodings for evolutionary design systems [5]. An interesting edited volume by Kumar and Bentley [6] provides an introduction to computational development and provides examples in the realm of robotics and neural network design.

When one considers approaches that are not restricted to combining growth and evolutionary computation for design purposes the field widens. For example, the use of evolutionary algorithms has been explored by various authors (e.g. [7–9, **?**,**?**]). Numerous applications of evolutionary computation to design are documented in the collections edited by Peter Bentley [10] and Bentley and David W. Corne [11].

3 Genr8's Algorithms

Genr8 is a plug-in for Alias|Wavefront's 3D design tool Maya and it was developed by the Emergent Design Group at MIT in 2001. More information can be found on `http://projects.csail.mit.edu/emergentDesign/genr8/`). The EDG is an interdiciplinary group that tries to develop new ideas in architecture by bringing together computer scientists and architects. A more technical description of Genr8's core growth and evolutionary algorithms can be found elsewhere [12].

3.1 Organic growth algorithm

At the heart of Genr8 is the growth algorithm for generating surfaces. The algorithm is based on Lindenmayer systems (L-systems) which have been widely and succesfully used to model plant growth [13]. An L-system is a *grammar*, consisting of a *seed* and a set of *production rules*, plus a *rewrite* process in which productions rules are repeatedly applied to the seed and its successive states. In its most stripped down form, an L-system can be considered as a system for rewriting strings of symbols. By combining them with a graphical interpretation of the generated strings, we have a powerful means of generating graphics. By far the most popular method of representing L-systems graphically is *turtle graphics*, where the symbols are interpreted as instructions for an imaginary turtle moving about in 3D space drawing lines. An L-system should be understood as a set of instructions for how to create a specific form rather than an exact blueprint detailing every coordinate. A specific characteristic of L-systems which is responsible for the organic appearance is that at each growth step the entire surface will be modified concurrently rather than by the sequential addition of elements.

A limitation of the basic L-systems model is that it can only create arboreal topologies. To generate surfaces one has to employ the *Map L-systems* algorithm [13]. In Genr8, the Map L-systems algorithm has been further extended and we

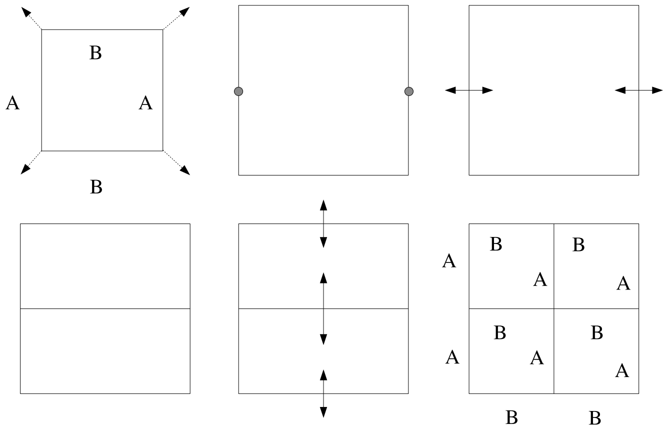

Fig. 1. One derivation step of a HEMLS rewrite system for creating symmetric squares. Each growth step has three phases. First the size of the surface is increased by moving the vertices away from the geometrical center of the surface. The square to the upper left is the seed and the arrows indicate how the vertices should be moved when increasing the size. In the second phase the rewrite rules are applied to each edge in the surface. In this example, the A edges are split and the new vertices are indicated by circles. Next, the branches are drawn and connected. The same procedure is applied to the B edges in the middle panel on the bottom. The figure on the bottom right shows the surface after one iteration of the rewrite rules.

use Hemberg Extended Map L-systems (HEMLS) to create surfaces in 3D which are grown in a reactive simulated physical environment. An example of a simple HEMLS grown in an empty environment is shown in Figure 1. A HEMLS requires the specification of a seed (or planar surface), a set of production rules and three additional parameters: we label this collection a rewrite system.

Genr8's environment is specified by the user and it has a significant impact on the outcome of the growth process. There are two types of elements in the environment, forces and boundaries. Forces can either be point-like attractors or repellors which act like magnets to make the surface grow towards or away from their location. There is also a gravitational force which uniformly directs the growth along one of the principal coordinate axes. The boundaries can either be placed as obstacles or used as bounding boxes to enclose the surface. Examples of the environment are shown in Figure 2.

3.2 Evolutionary computation

Creating a rewrite system that grows interesting surfaces by hand is a complicated task. This stems mainly from the difficulties of imagining what a given rewrite system will look like after repeated iterations. The influence of the environment only serves to exacerbate this problem. Additional complications arises from the difficulty of making sure that the rewrite system is syntactically correct.

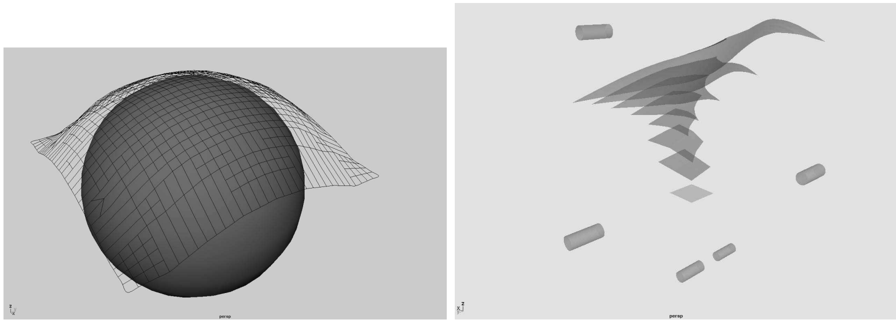

Fig. 2. The figure shows two examples of the square rewrite-system from the example in Figure 1 grown in two different environments. Left, the seed was placed above the sphere and pulled down by gravity as it grew. The surface was prevented from growing through the sphere and instead it drapes the sphere. Right, the surface is pushed upwards by the two repellors beneath it. During the development it was further deformed by the repellors (each of the five repellors are located at a single point, but drawn as cylinders). The figure shows all derivation steps overlaid and the shape of the seventh surface is far from the original flat square.

It is with these concerns in mind that an EA was incorporated into Genr8. The EA will automatically create valid rewrite rules and the user can exert high-level control over the process through the fitness function.

The particular type of EA used in Genr8 is called Grammatical Evolution (GE) [14]. The main advantage of GE is that it combines the strengths of Genetic Algorithms (GAs) [15] and Genetic Programming (GP) [16]. What makes GP such a powerful algorithm is that it provides for the evolution of executable structures represented as trees. For example, in Genr8 the executable structures are rewrite systems that are later grown in the simulated environment. Unfortunately, the genetic operations often become very complicated when one deals with subtrees that have to be compatible when swapped. GAs on the other are very convenient when one applies the genetic operators since the genome is represented as an array of integers as a representation of the genotype. The key invention in GE is to map a GA-style linear genome into a GP-like tree-structure. This is achieved by mapping the set of integers to the desired language via a Backus-Naur Form (BNF) representation of the grammar. This powerful technique can be applied to any language which can be represented by a context-free grammaar. All constraints of the language are handled by the BNF and GE thus provides a strict separation between the representaion of the genome and the target language.

Unlike many other EC applications, Genr8 has an additional mapping step. Once the genome has been mapped to a rewrite-system, it is grown in the simulated physical environment. As illustrated in Figure 2, the environment can have a significant impact on the outcome of the growth process. Once the surface has been grown, it is assigned a fitness value based on a number of distinct attributes. Consequently, changing the environment can lead to the surface attaining a different fitness value.

A crucial part of an EA is the fitness evalutation which guides the search towards better solutions. In design, there is no general way of algorithmically defining a 'good' surface. Coming up with a useful fitness evaluation scheme for design applications is still an open research question [17]. In Genr8 we use a fitness evaluation scheme that gives the user high level control of the evolutionary search. This has been implemented as a multi-parametric fitness function. Each parameter represents a specific feature of the surface. The user may set target values for each parameter as well as weights to determine the importance of each criteria. The total fitness is $F_{tot} = \sum w_i F_i$, where i runs over the six different criteria and the weight w_i is a non-negative real number indicating the relative imoprtance of each criterion. The six criteria are size (measures the extent in the x and y-directions), symmetry, soft boundaries (growing through walls allowed but penalizes fitness), subdivisions (measures how refined the surface is), smoothness and undulation (local and global measures of the variation in the z-direction).

It is important to point out that in most situations there are many different ways to attain a given fitness value. That is, the fitness function is degenerate in mapping the surfaces to a single fitness value. Consequently, there are many different surfaces which are equally good solutions for a given set of fitness criteria. This is an advantageous feature since it makes it easier to maintain a diverse population. Moreover, some of the criteria are more or less in conflict with each other. This means that the EA must negotiate a trade-off between the different criteria. These situations lead to the most interesting outcomes and help increasing the variability of the population.

4 Using Genr8

Genr8's use of growth and evolution yields a provocative and novel design process for an architect. Conceptually, the interaction and control between the architect and tool are radically changed. The designer has three means of controlling the design that emergently arises from interaction between computer and herself: 1) by setting up the digital physical environment 2) by supplying the growth instructions, and 3) by interactively guiding the evolutionary algorithm. In practical terms, the designer can interface with Genr8 either via a graphical user interface or the Maya's built-in scripting language, MEL (Maya Embedded Language).

The environment is set up before the growth is started, but can also be modified at a later stage between evolutionary generations. The designer has considerable freedom in specifying the environment. There are separate commands for creating attractors and repellors in the Maya scene. The attractors and repellors are represented as standard Maya cylinders and can be manipulated as standard Maya objects. In principle, any Maya object can be used as an obstacle or boundary. However, in practice, for boundaries the best results are achieved if they are smooth (such as regular polyhedra and spheres) and not placed too close to the seed.

Our experience has shown that designers are usually reluctant to give up any form of creative control to a computational tool. Thus, with respect to interactive control, our intent is that the overall design experience is analogous to the designer driving a car (in the driver's seat) where the evolutionary algorithm acts as engine. In terms of high level control, the car drives forward based on the designer-chosen parameter values of the fitness function. These values guide selection of fitter parents for creating the next generation of offspring. The designer uses Genr8's so-called "interruption, intervention and resumption" (IIR) control mode like the steering wheel, brakes and gas pedal. When the designer hits the brakes ("interrupt"), the current results can be inspected and the evolutionary process can be redirected ("intervene and resume"). In terms of inspection, the designer has flexible access to detailed information about the population. The rewrite system for each surface is available for replaying the growth steps and closer inspection. The surfaces' fitness are factored into the six different fitness criteria, making it possible to determine to what degree the different criteria contribute to the evaluation. There are also many options for redirection. For example, the designer can use standard Maya commands to investigate and modify surfaces. The computed fitness value of a surface can also be overwritten to emphatically select it more often because the designer subjectively prefers it very strongly. Moreover, any parameter such as mutation rate or the fitness function weights can be changed. Typically, to evolve a family of designs from one "run" of the evolutionary algorithm, the the designer periodically adjusts it and resumes for a few more generations.

In fact, Genr8 has a large array of parameters that can be adjusted by its designer. The drawback of this large degree of freedom is that it can be difficult for the designer to fully comprehend the meaning of all parameters and intuit how they relate to each other. We have observed that the most frequently used strategy is to hold most parameters constant and focus only on a exploring the possibilities provided by modifying a small set. The interactions of even a small set are sufficiently rich for creative design.

5 Genr8 projects

Genr8 has been used in student projects for the Emergent Design and Technologies (EmTech) program at the Architectural Association (AA) Inc, London, UK since 2003. Since it is available free on the Internet, it has also been picked up by architects worldwide. In this section we briefly describe 6 Genr8 projects.

Designer: Steven Fuchs
Project: Butterfly machines (2005)
Goal: To explore the implications of creating a family of designs under simple conditions that allow environmental influences to be directly interpretable.

Methodology: One advantage of parametric geenrative design is that one can easily obtain a whole family of designs with only minor variations by changing one or more parameters (see Figure 3). By incrementally changing just a few of the parameters, Genr8 enables a visual exploration of digital evolution. As a form finding experiment, the use of algorithmic scripting enhances the design process by allowing for iterative selection - an approach which mimics natural selection using an organic growth model.

Fig. 3. A family of Genr8 surfaces based on simple parametric variation of the square rewrite system.

The Butterfly machines project forgoes the EA and instead uses the predefined square rewrite system (see Figure 1). In an empty environment, the square rewrite system creates a symmetric surface. One of the advantages of using the square rewrite system is that the surfaces can be directly represented as NURBS surfaces (rather than the default polygons) which provides more opportunites in the post-processing steps. The forms created during the investigation were created in an environment with multiple attractors and repellors. In keeping with the design intention, each Genr8 run produced a self-intersecting surface as shown in Figure 4. In a later stage of the project an investigation was conducted into how one could panelize parametric surfaces based on material properties. The design intent was three-fold: panel size variation, material-based logic and scriptable optimization. Each panel is a flexible component placed in the context of an optimized point-cloud and carried by four input points. Using the geometric chord ratio and the ability of a material to bend, an optimization algorithm was created to tighten the enclosure of a double curved surface. The

Fig. 4. An example of a symmetric self-intersecting surface from the Butterfly Machines project.

solution sought to find a way to rationalize each surface with polygon quads in an attempt to control the light apertures and economize fabrication.

Designer: Achim Menges

General Intent: To utilize Genr8 to embed the possibilities and constraints of fabrication and assembly processes already in the computational form generation.

Project: Sectional Surfaces (2003)

Goal: To synthesize digitally evolved geometry and computer-aided fabrication processes through the definition of fitness criteria that embed manufacturing logics and material constraints as generative drivers in the morphogenetic process.

Methodology: Genr8 was employed to initiate the co-evolution of two interlocking surfaces with increasingly complex geometric articulation. The experiment commences from the possibility of describing the geometrical data of surfaces with varying curvature as systems of tangential and perpendicular construction planes interpreted as planar elements fabricated by computer-aided laser-cutting of sheet material. Thus a number of geometric constraints informed the definition of fitness criteria to ensure that the elements ended up in the correct planar fashion.

Project: Pneumatic Strawberry Bar (2003)

Goal: A design for a pneumatic strawberry bar for the Architectural Association's annual end of year party, intended to utilize the evolutionary dynamics of reproduction, mutation, competition and selection as design strategies.

Fig. 5. A photograph of the sectional surface model.

Methodology: The possibilities and limits from initial form generation to the actual fabrication process were explored by shifting the investigation towards performative patterns that evolve as species across populations and successive generations whilst maintaining structural capacity and geometric characteristics. The starting point of the Genr8 driven development process was a relatively simple pneumatic component geometrically defined by the cutpattern of two trapeziform surfaces that were aligned at the plane of the connecting seams. Once inflated the component becomes a three dimensional form defined by the different length of the surfaces in relation to the defining points. These simple geometric relations, defined as generic 3D cut patterns, provide the basis for the subsequent evolutionary process. Rather than breeding just one surface, three sub-populations were used in scheme based on co-evolution. A feedback loop was initiated where the most recently evolved surface was used as a bounding box for the current surface. This method maintained the inherent logic of the pneumatic component in a larger system but dissolved the distinction between environmental constraints and individual response. Another feedback loop utilised digital form-finding in a dedicated membrane engineering software, and additional physical test-modelling further informed the evolutionary process and its evaluation.

After running Genr8 for over 600 generations, 144 species were identified and catalogued according to specific patterns of relevant geometric features. Considering the interrelated evolution of the geometry-defining surfaces the criteria for evaluation was the relative fitness amongst the emergent species rather than the absolute fitness ranking of any particular individuals. Since the structural

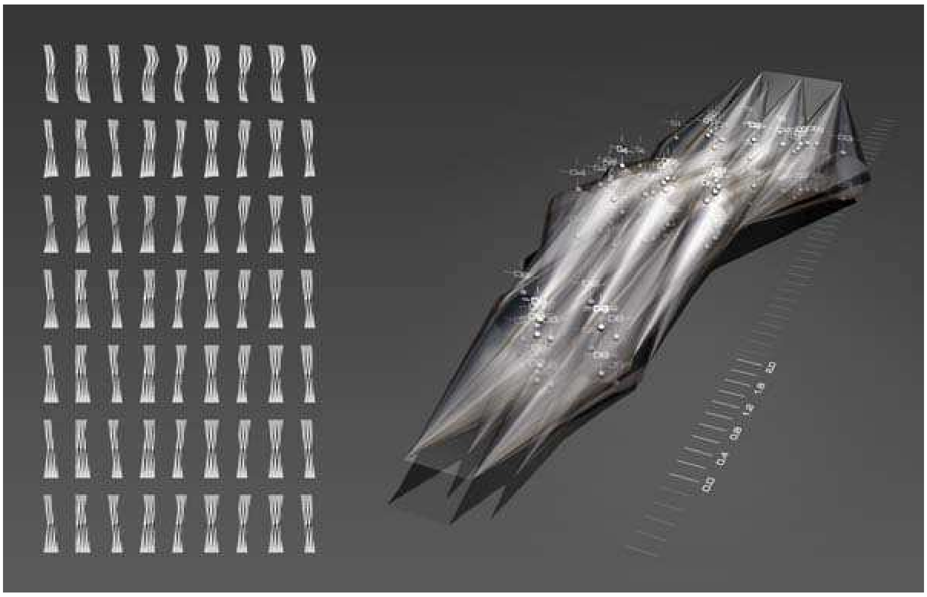

Fig. 6. Left, a selection of surfaces evolved for the strawberry bar. Right, a rendered model of the pneumatic strawberry bar.

behaviour of the pneumatic system relied primarily on specific geometric relations such as alignment and proportional distances of definition points, the individuals that shared these geometric features were selected. Then the individual of the chosen species that grew in the last and most developed generation was picked. The genotype of this individual incorporated the genomes of three geometry-defining surfaces, establishing a degree of phenotypic plasticity that allowed the resulting pneumatic system to adjust to the constraints of a digital cutting pattern and computer-aided manufacturing process.

Project: Fibrous Surfaces (2005)

Goal: To investigate possibilities of combining digital growth and associative parametric modelling to evolve a differentiated surface structure consisting of a dense network of interlocking members from a basic array of simple, straight elements.

Methodology: The basic system constituent is defined as a jagged, planar strip cut from sheet material on a three axis CNC router. First a generic digital component was established in a parametric software application through the geometric relationships that remain invariant in all its possible instances as well as the variable production constraints of the intended matchining technology and process. The use of this parametric component is then based on three interrelated inputs: Primary input influencing the particular geometry of a specific system type is given by a gestalt envelope. During the growth process of this

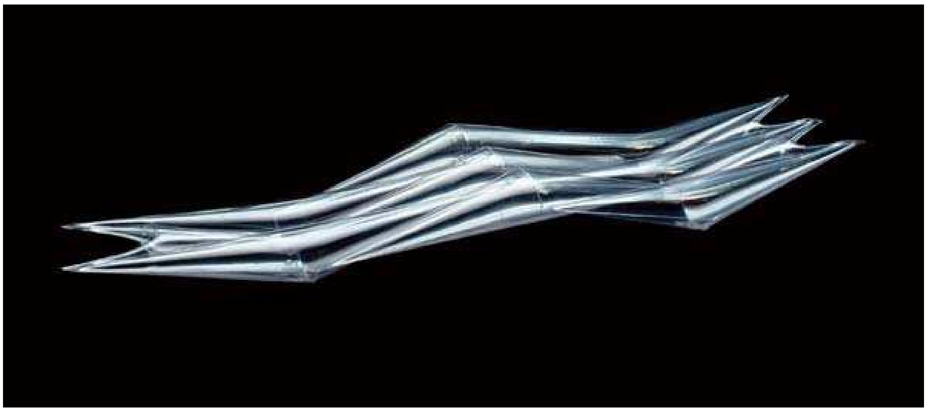

Fig. 7. A photgraph of the pneumatic strawberry bar.

envelope in Genr8 all parts of the surface continuously change until the ontogenetic drifts settle into a stable configuration. Based on the derived surface another input for the implementation of the material elements is generated. In response to particular geometric surface features derived through Genr8 a variable distribution-algorithm establishes a network of lines on the surface indicating the position of each element and the related node type. Instances of the generic parametric component then populate the system accordingly. In the resultant organisation crossing members only intersect if they are perpendicular due to the embedded manufacturing constraints. If not, they pass under or over crossing elements, not dissimilar to a birds nest, and thereby form a geometrically defined, self-interlocking, stable structure. This complex correlation of geometric definition, structural behaviour and production logics does not only remain coherent in a single system, but it is integral to the generation process itself. This is of particular importance if one considers that the surface defining the critical morphogenetic input is constructed through a bottom-up process in which all parts respond to local interactions and the environment. As these internal and external interactions are complex and the interpretation of the L-systems is non-linear, the outcome of the growth process remains open-ended. This continual change combined with the long chain dependencies of the subsequent parametric componenet population enables the growth of different system types of member organisation, system topology and consequent performative capacity.

Designer: Katrin Jonas

Project: Surface envelopes, (2003)

Goal: As part of a general investigation into using surfaces to define inhabitable spaces, the objective of this project was to explore how external conditions would influence the growth of the surfaces that would create a covered space underneath.

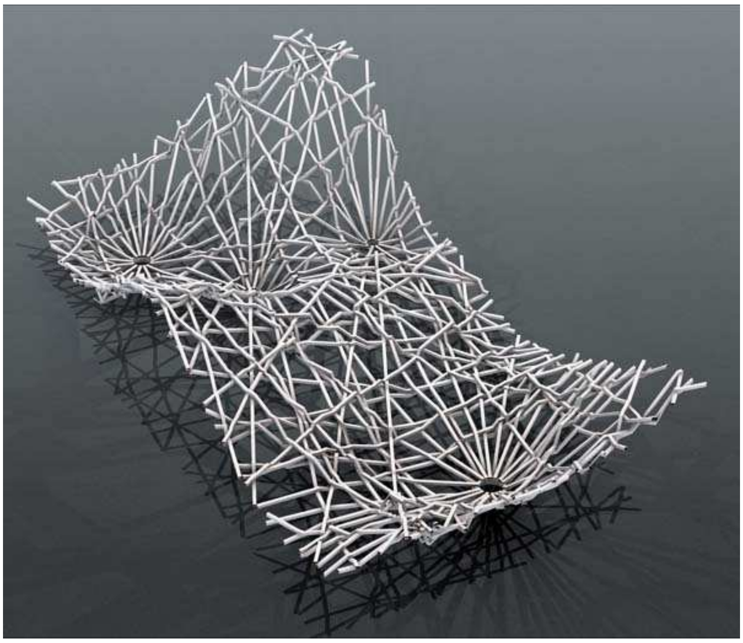

Fig. 8. A view of the fibrouos surface model. This prototype has almost 90 members and 1000 joints

Methodology: The environment as shown in Figure 9 included a spherical obstacle, two attractors on a lower level than the starting point and a bounding box. The size of the bounding box was made smaller than the surfaces so that the surfaces would collide with it and expand along the walls, creating a rim of overlapping geometries.

The set up allowed for a clear definition of a design problem and it also allowed for preconceived ideas on what the outcome could look like. The expected abstracts of surfaces which describe a path from a higher to a lower level in space were challenged by the multitude of outcomes that actually evolved. The surfaces which evolved in this environmental setup responded in an unexpected but nevertheless understandable way. Where the sphere was hampering the growth, the surfaces would yield and expand upwards. The attractors produced a number of different outcomes: some surfaces grew downwards as was expected, others split into two branches before descending with each branch reaching into the direction of one of the attractors while others did not drop after splitting into two branches.

The next step was to select a number of surfaces for further analysis and manufacturing. Here the fitness function proved useful as the system produced several distinct solutions with similar fitness values. Despite their distinction in the arrangement of geometric elements, all surfaces shared the common feature of sharp edges. One of the problems with the forms however was that there

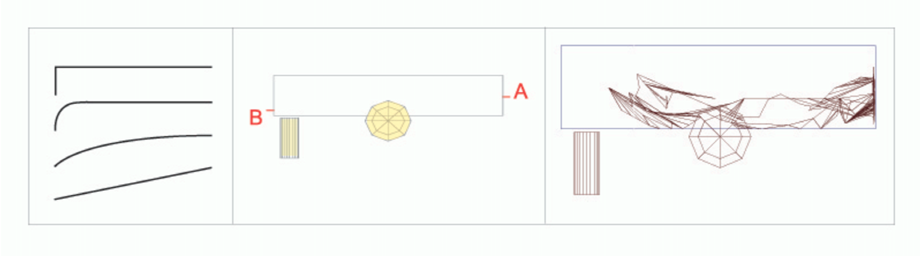

Fig. 9. Left, the conceptual sketch for the environment prior to using Genr8 for the Surface Envelopes. Center, elevation view of the environment used in Maya, the cylinder at the bottom left represent two attractors (one can not be seen as it is placed directly behind the other), a sphere and a bounding box. Rigth, an example of an evolved surface from the experiment.

was no definition of the centre part of the surfaces when shading them. The individuals all just described a definite rim. In preparation of the computer aided manufacturing of the design, focus was placed on defining the corpus of the forms. This was achieved by triangulating and post-processing the output in a number of different software applications as shown in Figure 10. Through this procedure, a cutting pattern that could be supplied to a laser cutter was obtained.

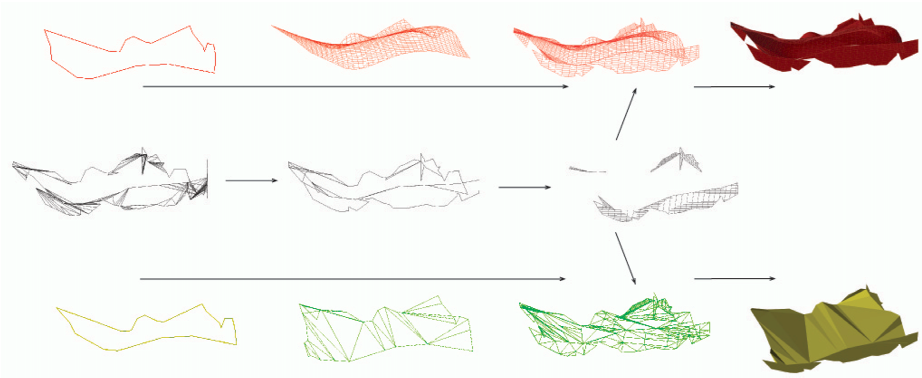

Fig. 10. The figure illustrates how fundamentally different objects can be produced through different methods of interpreting a particular surface outline. The middle row shows the processing of the initial Genr8 rim, in that geometries which overlay each other were filtered to share a single outline. The upper and lower rows each show the development of a central body definition. Starting from the rim outline in the upper sequence a smooth definition is applied and in the lower row a folded definition. Subsequently the filtered outrigger of the rim are reattached.

For the manufacturing, aluminium was chosen since it allows for an uncomplicated manual folding process. The metal sheets can be folded along score lines without deforming the faces. After processing the pattern, the sheets were folded up into the final physical objects. Each model as shown in Figure 11 represented one possible interpretation of a Genr8 surface.

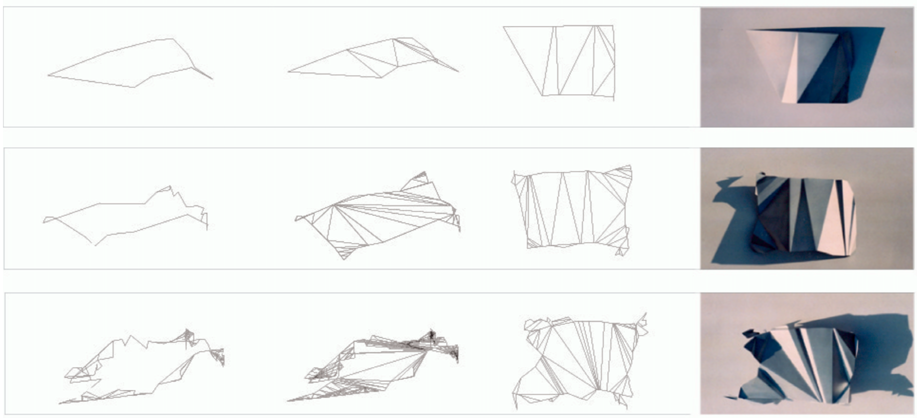

Fig. 11. Three different Genr8 surfaces with the same fitness values but different degrees of complexity in their articulation. The surfaces are triangulated and unfolded to create a pattern which can be used for laser cutting and scoring. The first column shows the outlines obtained from the Genr8 surfaces and the second the triangualtion which obtained using Maya. The third column shows the unfolded version and finally there are photographs of the produced surfaces.

Designer: Michel da Costa Gonçalves
Project: Nested Cubes, (2004)
Goal: To investigate the use of Genr8 as a semi-automated spatial sketching tool and to consider how Genr8 can propose contrasting hands-off "design solutions" that inhabit a particular environment. This will test the opposing principles of convergence and diversity which are inherent in an evolutionary algorithm.
Methodology: The project used a literal illustration of inhabitable constraints such as overall limitation and internal desirable layout of spaces. The graphic interpretation is taken as a direct representation of a material envelope. The skin-like surfaces are interpreted as a spatial enclosures filling a physical setting. This environment defines both an absolute boundary and nested obstacles. The external boundary is represented by a cube containing up to three smaller cubes as shown in Figure 12. A script was developed to automatically set up the environment and assigning random positions to the inner cubes. There scene

also included repellors pushing that would push the surfaces up or down, forcing them to interact with the obstacles.

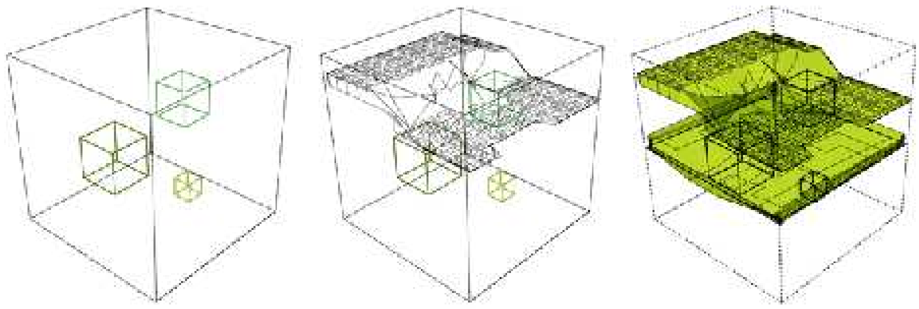

Fig. 12. The leftmost image shows an example of the environments used in the Nested Cubes project. The middle image shows an upper surface and the last image shows the upper and the lower evolved surfaces.

Due to the absence of direct representation of structural constraints, the aim was oriented towards spatial consideration. The obstacles' positions were representative of a given architectural layout while fitness parameters simulated inhabitable characteristics. Spatial configurations were generated by filling the environment in accordance to its embedded criteria. Fitness was prioritized towards local continuity; giving weight to smoothness criteria. The selection criteria also included a balance between local variations of the "envelope" and degrees of conformity to a given configuration.

The next step was to tune the range of parameters that could be used in the project. For the growth algorithm, this meant establishing a scale ratio between the surface cell and the overall environment. For the EA the goal was to relationship between Genr8's fitness criteria and the recurrent features of the environment. As shown in Figure 12, an upper and a lower surface was created for each configuration.

Automated loops of incremental of the parameters made it possible to generate 1008 configurations and 168 "optimised" enclosures grouped in comparative charts used to identify these dependences. In addition to fitness criteria, the tuning phase studied the growth algorithm by establishing the scale ratio between the surface cell and the overall environment. The results were reviewed according to modes and degrees of interactions such as the proportional relation between cell and frame, surface coverage and spatial conformity. The process included discarded conditions where the surfaces displayed limited occupancy or overcame obstacles and boundaries.

The results were exported, rebuilt and transformed in different manners in order to recreate the continuity between the surfaces following the boundary limits. For example, one operation sought spatial coherence by locally increasing the curvature continuity of separate contours. After being reconstructed and lofted, these profiles generated a seamless envelope. Successively, the same conditions were rebuilt with a regular meshing related to a proportional structural meshing.

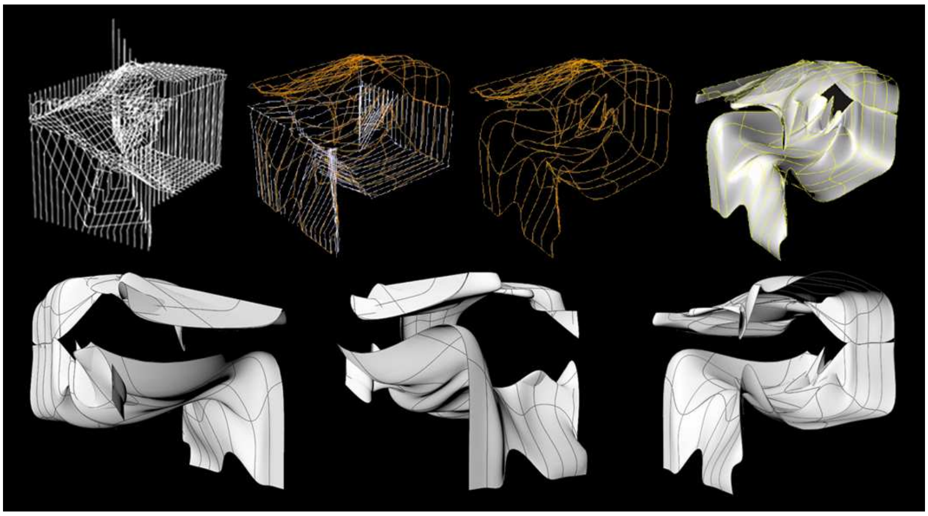

Fig. 13. An illustration of the sequences of transformations of the evolved surfaces from Figure 12. Through a number of different operations, a smooth spatial envelop is created.

This accessible interface expresses basic architectural requirements while using predefined Genr8 settings to provide envelope solutions generated according to spatial criteria. This process could be largely extended by giving more control over the range of fitness values in relation to design criteria. This project illustrates an abstract transcription of design variables different in nature and scale. It includes different levels of "external" constraints influencing the surfaces' generation combining the intangible criteria guiding the generation/creation with fixed geometrical/spatial layout. By domesticating seemingly unpredictable tools, it offers consideration for added design characteristics; dissociated generations of fitted results. This requires the ability to manage the equilibrium between converging fitness and extensive generations. By introducing redundancies through the abundance of generation it exhorts a clearer ability for defining modes of selection. This hints to the difference between Genr8 and other digital design tools; the dissociation between the user of form generation and the seemingly unpredictable character of the procedure. Design expertise, and to a certain extent, creativity could be reconsidered towards the integration of nonlinear formal generation within a design process.

6 Discussion

The designers agree that Genr8 is different from any other design tool they have used and that it presents them with new possibilities as well as new challenges. Genr8 uniquely offers the designer a chance to engage in a non-linear generative bottom up design process. However, in order to exploit this opportunity, the designer must relinquish some conventional aspects regarding her role. With Genr8 the control of the design process is shifted slightly out of the hands of the designer. The designer must both accept the fact that she no longer has complete control and adjust to new ways to exert control. The new mode of control is indirect: it is through the definition of the environment, the specification of parameters and interactive guidance of the evolutionary selection process.

The developers' aim was to develop a design tool that would empower designers who had only rudimentary knowledge of evolutionary computation and L-systems. However, the designers found it challenging to understand the distinction between the behaviour of the growth algorithm and the behaviour of the evolutionary algorithm while they observed that this is necessary to exploit the tool to its greatest potential. A designer using Genr8 does not have to understand the algorithmic details within it but she must have a coherent and sufficiently accurate grasp of the tool's behavior. An inaccurate mental model creates a frustrating gap between desired and actual outcomes. Over time, especially through teaching at the AA by Hemberg and Menges, improved explanations of Genr8 that enable better mental models have been formulated. This chapter is such an example.

Since both the evolutionary and growth algorithms influence the complex output that arises as a result of reacting to the environment, it is initially difficult to disentangle their individual roles in the output. There is a learning process during initial experimentation when the designer comes to appreciate the separate aspects of the two algorithms. This sets up a considerable tool adoption challenge: to figure out how to best exploit Genr8 toward accomplishing a set of goals. To do this, the designer's task becomes one of figuring out how to express her design criteria in ways that are amenable to Genr8's framework. Since so many new and unorthodox concepts are incorporated into Genr8, this task is reported to often be non-intuitive. Genr8 is sometimes unwieldly. The challenge for the designer lies in combining and understanding the abstract parameters and logics of the evolutionary algorithm and the growth algorithm with the geometrical and spatial layout at hand. The environment fortunately is much more intuitive to work with. It provides a powerful means of representing a wide range of influences, both physical and more conceptual notions. The designer must learn to appreciate the solutions proposed by Genr8 and determine to what extent the tool has managed to negotiate between the different constraints and performance criteria.

Genr8 was originally conceived as a sketching tool used in an early stage in the design process. It was predicted to be useful for deriving broad conceptualizations of form that would subsequently undergo more detailed definition and analysis with respect to structure or material. It was expected that the lack of structural and material analysis in Genr8 would limit its specific value. Although the environment can be set up to reflect the physical reality to a certain extent, the fitness criteria and the parameters are inherently geometric. This forces the designer to interpret the structural or material constraints in unwieldy geometric terms (e.g. by restraining certain angles or the distances between support points). By interacting with the designers using Genr8 we have learnt that it is powerful to a greater extent than its developers may have conceived. Architects can accelerate it with unanticipated techniques and solutions that surmount its perceived limitations regarding structure and material considerations.

The overall scenario of early stage use was indeed adopted by the designers. Nevertheless, the designers did not perceive geometry to be as big of an obstacle as originally feared. Instead they considered Genr8's approach to creating surfaces through a generative growth algorithm to be an opportunity. In contrast to most contemporary design, Genr8's approach is not based on the artificial distinction between processes of form definition and making. Instead it derives morphology from the inherent constraints and possibilities of production and construction. Thus Genr8 becomes not only an enabling software tool but also a strategic vehicle for understanding and instrumenting the design process as truly morphogenetic, in which process formation and materialization are always inherently and inseparable related.

While a population based search method that produces a *family* of designs is novel, it can be embraced and return highly interesting and exciting options. For example, a designer can obtain surfaces which are equally well adapted to the given fitness criteria, but nevertheless quite distinct as illustrated in Figure 11.

At the outset of Genr8's development in 2001, the developers had many goals: provide a proof of concept that ideas from evolutionary computation and Artifical Life could be useful in architectural design, address the complicated issue of developing natural shapes with biologically inspired computational process, and develop novel software that is truly creative and can actively assist the designer in coming up with new ideas and suggestions. Through collaborative dialogue both the designers and Genr8's developers are in agreement that Genr8 has achieved its goals.

7 Summary

We have developed an open ended creative design tool which uses an organic growth model and an evolutionary algorithm. The surfaces are grown in a parallell fashion with all parts changing and expanding at the same time during each growth step. This type of processes is hard for a human designer to replicate and the resulting shapes has different quality from what most designers achieve by hand. To achieve this, we have extended map L-systems in such a way that we can create surfaces in 3D space. Genr8 uses Grammatical Evolution to evolve rewrite systems which are interpreted to form surfaces. The fitness function is a weighted combination of specific surface features. Genr8 is interactive and attempts maximize the possibilities for its designers to maintain creative control.

The paradox of making Genr8 open ended is that, while it does not constrain the designers who use it, predicting how it would be used was impossible. Through observing Genr8 in use it has at least been possible now to elucidate some strategies that successfully exploit it. Yet, we remain certain that not all of its possibilities and potential have been enumerated to date. If Genr8 continues to be disseminated in a way wherein its adopters must figure out how to use it by themselves, it will continue to be integral to many creative and inventive design interactions.

Acknowledgements

We would like to thank Peter Testa, Devyn Weiser, Simon Greenwold and all the other members of the EDG at MIT for their contributions to developing Genr8. We are also grateful to Michael Hensel and Mike Weinstock at the AA for stimulating discussions. Finally, we would like to thank all the students and other users that have provided valuable feedback over the years.

References

1. Thompson, D.: On growth and form. Cambridge University Press (1961)
2. Frazer, J.: An Evolutionary Architecture. Architectural Association, London (1995)
3. von Neumann, J.: The theory of self-reproducing automata. University of Illinois Press (1966)
4. Broughton, T., Coates, P., Jackson, H.: Exploring 3d design worlds using lindenmayer systems and genetic programming. In Bentley, P.J., ed.: Evolutionary Design by Computers. Morgan Kaufmann (1999)
5. Hornby, G.S., Pollack, J.B.: The advantages of generative grammatical encodings for physical design. In: Congress on Evolutionary Computation. (2001)
6. Kumar, S., Bentley, P.J., eds.: On growth, form and computers. Elsevier (2003)
7. Murawski, K., Arciszewski, T., Jong, K.A.D.: Evolutionary computation in structural design. Eng. Comput. (Lond.) **16**(3-4) (2000) 275–286
8. Shi, X.G., Gero, J.S.: Design families and design individuals. Eng. Comput. (Lond.) **16**(3-4) (2000) 253–263
9. Maher, M.L.: A model of co-evolutionary design. Eng. Comput. (Lond.) **16**(3-4) (2000) 195–208
10. Bentley, P.J., ed.: Evolutionary Design by Computers. Morgan Kaufmann (1999)
11. Bentley, P., Corne, D., eds.: Creative Evolutionary Systems. Morgan Kaufmann (2001)
12. Hemberg, M., O'Reilly, U.M.: Integrating generative growth and evolutionary computation for form exploration. (Submitted to Genetic Programming and Evolvable Machines)
13. Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Springer (1991)

14. O'Neill, M., Ryan, C.: Grammatical Evolution - Evolving programs in an arbitrary language. Kluwer Academic Publishers (2003)
15. Mitchell, M.: An introduction to genetic algorithms. MIT Press (1996)
16. Koza, J.R.: Genetic programming: on the programming of computers by means of natural selection. MIT Press (1992)
17. Romero, J., Machado, P., Santos, A., Cardoso, A.: On the development of critics in evolutionary computaion artists. In: EvoMusart workshop, 6th European Conference on Genetic Programming, Essex (2003)