# Automated Shape Design by Grammatical Evolution

Manuel Muehlbauer[1(✉)], Jane Burry[1], and Andy Song[2]

[1] Spatial Information Architecture Laboratory (SIAL),
RMIT University Melbourne, Carlton 3000, Australia
{manuel.muehlbauer,jane.burry}@rmit.edu.au

[2] Evolutionary Computation and Machine Learning Group,
RMIT University Melbourne, Carlton 3000, Australia
andy.song@rmit.edu.au
http://www.sial.rmit.edu.au/
https://titan.csit.rmit.edu.au/e46507/ecml/

**Abstract.** This paper proposes a automated shape generation methodology based on grammatical genetic programming for specific design cases. Two cases of the shape generation are presented: architectural envelope design and facade design. Through the described experiments, the applicability of this evolutionary method for design applications is showcased. Through this study it can be seen that automated shape generation by grammatical evolution offers a huge potential for the development of performance-based creative systems.

**Keywords:** Shape generation · Design · Genetic programming

## 1 Introduction

This research presents a Genetic Programming (GP) approach to shape design using grammatical evolution to provide a methodology to extend the potential for the use of machine intelligence during early stages of architectural design processes. A method of this kind could greatly enhance efficiency and support decision making for architects. The opportunities and limitations of the proposed grammatical evolution for the design of architectural spaces is elaborated.

The main contribution of the presented research is the development of a methodology for grammatical evolution and the definition of the representation for the encoding of particular design spaces. It explores the potential of this process for the generation of simple envelopes - the exterior hull of a building, for architectural shapes, combined shapes for architectural envelopes and complex shapes for facade components.

During our research, different methods are used to develop a methodology for shape design by grammatical evolution. Strongly-typed tree-based GP [13] is used as the core. Shape-based nodes are used as main representations for the different design cases.

J. Correia et al. (Eds.): EvoMUSART 2017, LNCS 10198, pp. 217–229, 2017.
DOI: 10.1007/978-3-319-55750-2_15

The paper reports the research outcomes which are architectural shapes produced with gradually increasing complexity during the development of case specific representations. During the process of developing representations for architectural shapes, the feasible of those representations are explored and discussed in the experiments section (Sect. 4) and discussions section (Sect. 5) of this paper. Relevant existing studies from the literature are presented in the related work section (Sect. 6).

Knowledge generated throughout the different experiments presented in this paper is reported in the conclusion section (Sect. 7) and reflected upon in respect to the further development of the methodology in the future research section (Sect. 8).

## 2 The Need

During architectural computational design processes, usually a parametric model is defined to specify a particular architectural shape space after an intuitive design process has taken place. This parametric model is then evaluated for potential parameters that can provide the flexibility for optimization of the architectural shape to increase the performance of the architectural shape. As a result of this post-rationalization process, the design space for architectural shape is already limited to a small scope of design solutions that reflect the design decisions taken during the design process.

Shape design by grammatical evolution opens up the opportunity for architectural shape to incorporate performance criteria early on in the design process, increasing the impact of the shape generation on the performance potential. Therefore, the design space in this experimental process needs to encode a wide range of design solutions to effectively search for a high performing solution. Shapes generated throughout this evolutionary process can then be used to further inform the design process based on local and site specific criteria.

## 3 Methodology

The methodology of this research project can be divided in two parts: the grammatical evolution as shape generation process and the representations developed to encode a particular set of shapes that are of experimental interest to the field of architecture. The first will be discussed in this section, before greater detail about the implications of the process is provided in the discussion section. The latter will be described as part of grammatical evolution and in greater detail during the section concerned with the implementation and experiments.

### 3.1 Grammatical Evolution

Grammatical evolution is an approach to evolutionary computation that introduces genotype-phenotype mapping based on grammatical rules [17]. Search and solution space are often separated. This separation and the translation process allow the search process to iterate all possible genotypical expressions, while ensuring the validity of the grammar output [14]. Through this approach, a particular set of shapes can be generated and uniformity of the phenotypical expression guaranteed based on the translation through the shape grammar.

We present automated shape design by grammatical evolution successively in the next sections based on the components, as they are defined during the establishment of the process, starting from the initialization of the first generation.

### 3.2 Representations

A set of terminals and functions is provided for different particular design cases based on shape grammar. During this process, a set of terminals and functions is developed for each particular case. The genotype as a result of the generation is a set of integer values, structured through the progression of data throughout the data tree. The representation is tree-based and the root node is used to collect the different shapes in a coherent representation of the solution. These terminal and function sets are reported in the implementation section alongside the different experiments and the resulting sets of shapes that were generated based on these case specific representations.

### 3.3 Fitness Measure

Decision parameters are defined based on the design criteria for the architectural shapes and a fitness function is developed to allow the constant evaluation of the criteria during the evolutionary process.

### 3.4 Shape Grammar

During the development of the genotype, a set of grammar rules is applied to develop a phenotypical expression of the architectural shapes that can be evaluated towards the fitness functions. In this research, the objects for evaluation are polygon meshes generated through the application of different algorithms that translate the phenotype consisting of points and/or lines into objects.

Crossover and mutation are powerful operators in GP. When shape grammar is involved, we need to modify these genetic operators so specific grammar restrictions are applied to specify the shape for the design process. During this design process the representation is limited by a coherent grammar structure, while the parameters are flexible to explore the design space during the process. Constraints were integrated into the nodes that generate integer or float values through the definition of domains that allow specific shapes to emerge.

## 4 Implementation and Experiments

The implementation based on the above methodology and the corresponding experiments are detailed in this section. Note some implementation details are combined with the description of experiment as different parts of experiments involve different implementations.

### 4.1 Implementation

A variety of tools were used to ensure the accessibility of the process to architectural designers. The main environment used to generate architectural shape was Rhinoceros CAD which supports Rhinoscript, a scripting language for the 3d computer-aided design. It provides a set of specialized methods and functions for the design of complex geometries. Apart from that, a Python-based GP library was developed under Rhino to facilitate grammatical evolution. This gives designers the possibility to modify the process and adjust it towards their own needs during the design process.

**Genetic Programming:** Our genetic programming implementation utilizes treebased strongly-typed GP. A random initial generation is used as the starting point for the evolutionary process. In the fitness evaluation quantitative properties, of the geometry are used to evaluate individual solutions, in particular shape volume and boundary surface area.

During the evolutionary process, tournament selection (tournament size = 3) is applied as a selection strategy to provide a mating pool (mating pool size = 10) for the crossover operation (crossover rate = 0.6). Besides this selection mechanism, a number of individuals are selected for the modification through a mutation operation (mutation rate = 0.3) to enhance diversity in the population. The best performing individuals of a generation are preserved through elitism (elite rate = 0.1) to ensure a steady advance in performance of the best individuals.

During the experiments a small population of individuals (population count = 500) and a low number of generations (maximal number of generations = 100) are used.

**Shape Grammar:** After generating the genotype with its associated parameters during grammatical evolution, a phenotypical representation of the design is produced for evaluation. The presented experiments differ in terms of the representations as well as in the translation process from genotype to phenotype. The next section will describe these differences.

### 4.2 Experiments

A set of experiments has been designed for exploring the feasibility of abstract shapes as initial stage during architectural design processes. As massing is crucial for the design of architectural shapes, auto-generated shape as a starting point for further design opportunities opens a huge potential for reflection on the performance of building envelopes. The pre-optimization of architectural shapes allows a more abstract evaluation of shapes towards performance goals at a stage of the design, when major design decisions are yet to be taken. Therefore this experimental process allows a stronger focus on building performance and furthermore the information of the design process through the adaptation of design features with major impact on the performance of the architectural shape.

Three experiments were undertaken to explore this potential with increasing complexity of the geometry. The experimental process starts with simple shapes for architectural envelopes. The same approach is then extended to the combination of shapes to generate another set of design solutions. In a third experiment, a different shape generation process is introduced to explore the potential application of the process to the generation of complex shapes for facade components.

**I. Simple Shapes for Architectural Envelopes:** In this experiment, the representation of the architectural envelope is established through the collection of three integer nodes in point nodes to generate three dimensional point clouds. The shape nodes in this case consist of ten points that define the shape of the architectural envelope.

This genotype is then used as input for a convex hull algorithm that translates the point cloud into a polygonal mesh. The fitness function in this case is maximizing the volume of the shape, while minimizing the area of the polygon mesh, leading to compact shapes for the architectural envelope.

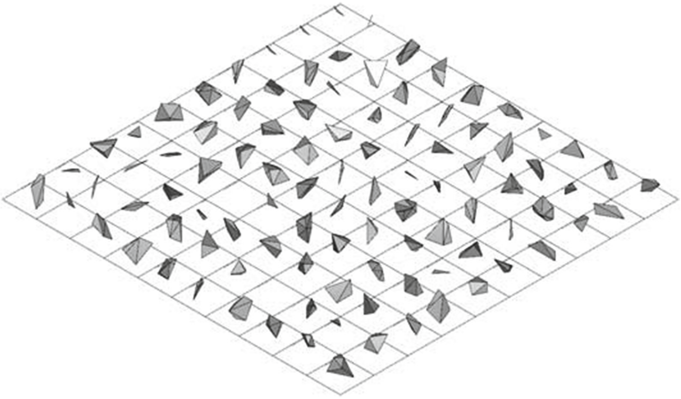

**Fig. 1.** Shapes from Experiment I - isometric view

$$\begin{aligned}\langle \mathit{Shape} \rangle \ &::= \ \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle,\\ &\quad\ \ \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle\\ \langle \mathit{Point} \rangle \ &::= \ \langle X \rangle, \langle Y \rangle, \langle Z \rangle\end{aligned}$$

$$Fitness = Volume/Surface\ Area \tag{1}$$

These shapes, as shown in Fig. 1, are providing solutions that reduce the surface area of the envelope to minimize thermal loss, while providing a large volume of space for the incorporation of architectural functions as the spatial program gets introduced to the project in the next design stage.

**II. Combined Shapes for Architectural Envelopes:** In the second experiment, the root node collects two different sets of points that define architectural shapes. Through this process, a new variation of architectural shapes is generated. These shapes are also consisting of ten points each that are defined by a collection of three integer nodes each.

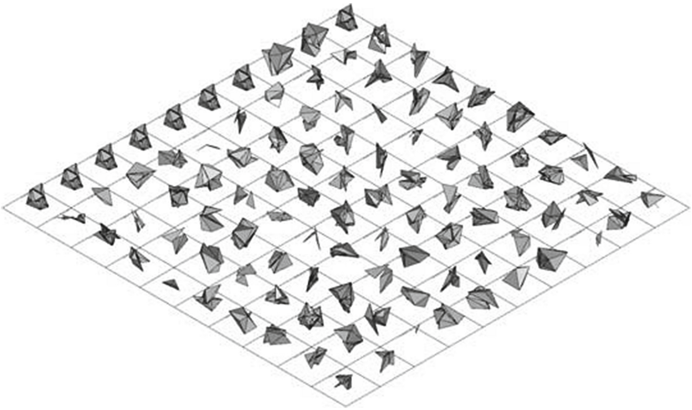

**Fig. 2.** Shapes from Experiment II - isometric view

$$\langle \mathit{Combined\ Shape} \rangle ::= \langle \mathit{Shape} \rangle, \langle \mathit{Shape} \rangle$$

$$\begin{aligned}\langle \mathit{Shape} \rangle ::= {} & \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \\ & \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle, \langle \mathit{Point} \rangle\end{aligned}$$

$$\langle \mathit{Point} \rangle ::= \langle X \rangle, \langle Y \rangle, \langle Z \rangle$$

The fitness function here is different from Equation (1). It is modified to minimize the enclosed volume, while the surrounding surface area is maximized during the grammatical evolution.

$$Fitness = Surface\ Area/Volume \tag{2}$$

Figure 2 presents the architectural shapes from this experiment. They have different advantages in the context of early design stages. Through the increase in surface area, the interface with the environment is maximized, while the volume for the architectural program is comparatively small. These shapes are feasible for design solutions with a highly differentiated architectural program with specialized spaces. Two application cases are relevant in this context. One is the introduction of collector surfaces for the use of solar radiation with either photovoltaic systems or algae reactors to generate biomass. The second is the use of the envelope for the display of information through the introduction of media facades.

**III. Complex Shapes for Facade Components:** The third experiment explores the application of grammatical evolution for the generation of facade components. This application case strongly deviates from the other experiments, so that the extend of the design space in z direction is restricted to refer to the two-dimensional character of facade panels. In the tree-structure of the representation, again ten points are collected based on three integer nodes. The domains of these integer nodes are differentiated to incorporate the restrictions in z direction, while the integer nodes for the x and y direction allow values between 0 and 500 mm, the domain set for the integer values of the z direction is between 0 and 100 mm. Another major difference to the previously described experiments is the use of a Delaunay mesh algorithm for the triangulation of the facade panels. Based on the hexagonal geometry, a grammar check is introduced that omits points that are outside of the domain to ensure consistency of the mesh geometry generated. Porosity is introduced to the mesh through the use of a mesh frame algorithm and the resulting meshes are thickened to provide material thickness. Through the smoothing of the shapes an organic aesthetic expression of the geometry is achieved.

$$\begin{aligned}\langle \mathit{Shape}\rangle &::= \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle,\\ &\phantom{::=}\ \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle, \langle \mathit{Point}\rangle\end{aligned}$$

$$\langle \mathit{Point}\rangle ::= \langle X\rangle, \langle Y\rangle, \langle Z\rangle$$

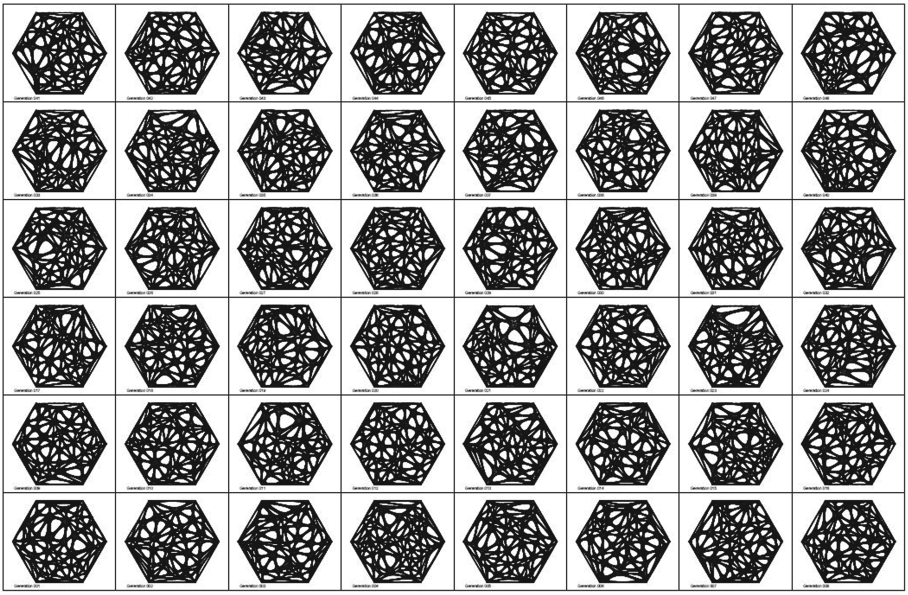

**Fig. 3.** Shapes from Experiment III - facade generation

Fitness evaluation for this design case is minimizing the volume of the geometry and therefore the material needed for the production of the facade panel and maximizing the area of the polygon mesh.

$$Fitness = Surface\ Area/Volume \tag{3}$$

The shapes generated in this final experiment that is reported in this paper are presented in Fig. 3. Through their properties, they suggest an application potential for the introduction of thermal mass in the building envelope that exhibits a high rate of heat exchange through the high surface-mass ratio. Another potential application of these geometries is the introduction of evaporative cooling through the material properties, as the fast heating of the surface will enhance these effects.

## 5 Discussions

In this paper the approaches to shape generation were presented and discussed for their application potential. During the implementation of the grammatical evolution different opportunities and challenges were encountered and will be reflected upon in this section.

### 5.1 Opportunities and Challenges of Grammatical Evolution

The experiments revealed the potential of grammatical evolution for the generation of abstract case specific architectural shape in three different approaches. The representations developed based on shape, point and integer nodes allow the definition of a design space that is general enough to evaluate possible shapes for a wide range of design cases, while specifying the design space for a particular design situation.

Grammatical evolution proved to be an appropriate methodology for shape design in the context of performance-based architectural design. It allows the generation of a wide range of feasible designs for the optimization of architectural shapes towards a particular set of criteria.

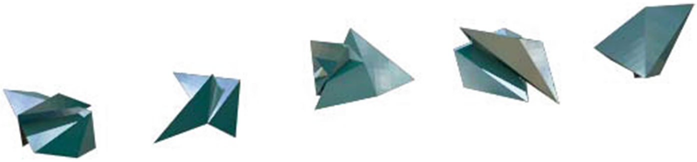

**Fig. 4.** Post-processed set of combined shapes - perspective view

The challenges are mainly in the implementation level. For example the development of the terminal and function set needs to be traded of against the performance of the architectural shapes generated during the process. Restriction of the integer domains and checking of feasibility of the point clouds in the context of complex shapes for facade components is another challenge.

### 5.2 Post Processing

Post processing of the generated shapes shows the possible extension of the design space. The use of texture and color in the design process widens the scope for grammatical evolution dramatically. Even if not integrated into the evolutionary process as yet, we want to present initial results for the use of these concepts in the following Fig. 4.

In Fig. 4 the set of combined shapes for architectural envelopes is presented with initial shading to show some of the design potential that is inherent in the shapes generated through the experiments. The use of the paneling process for the use in architectural design processes was tested during the development of the Hoverport as part of an architectural competition for the future of airports. Figure 5 illustrates our entry which is based on the auto-generated facade components described in Experiment III.

### 5.3 Composition

In this research a variety of simple shapes are generated and the potential of the combination of shapes to generate a variety of architectural envelopes are presented. In general, through the developed process, complex shapes can be generated for a variety of architectural compositions during the architectural design process. This potential for composition needs further exploration to reveal the full capacity of the process.

### 5.4 Flexible Representation

The flexible representation used in grammatical evolution contributed to the development of the architectural shapes through the provision of a certain design

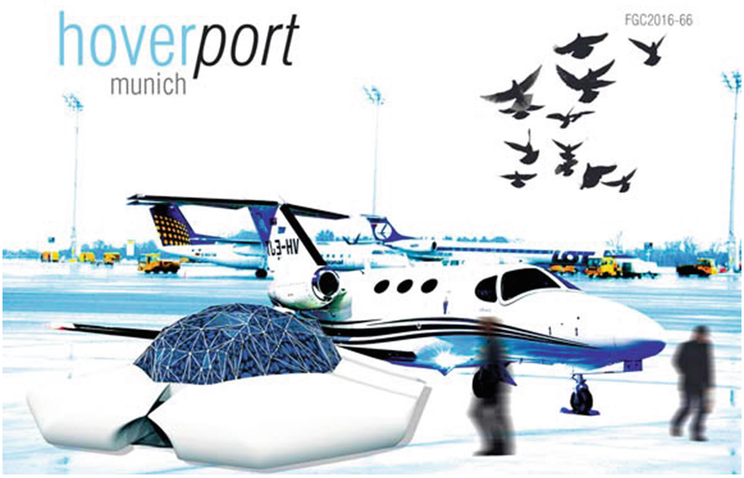

**Fig. 5.** Application of facade components in hoverport design

space that could then be systematically extended through the introduction of different shape grammar rules. This flexibility exceeds the current potential used in architectural design processes based on genetic algorithms and conventional search methods.

The presented research established a methodology for the use of grammatical evolution for shape design in architecture and reported on different experiments to generate shapes through the development of case specific representations using shape grammars. Based on the grammatical representation of the architectural shapes, grammatical evolution was introduced based on strongly-typed tree-based genetic programming. This approach allows the generation of a variety of shapes based on flexible representations.

## 6 Related Work

In this section the background of the research is discussed based on key references and the research is positioned in the context of related work in the field of shape grammar and grammatical evolution. The implementation strategy used for the grammatical evolution is based on the work of Lee, Herawan and Noraziah [11]. As Lees research is focused on the evolution of industrial products, the presented research is set apart through the development of representations for architectural shapes.

The line of research using shape grammar in architectural design originated with Stiny and Mitchell in 1978 [18] through the development of the Palladian grammar. Following on, Koning and Eizenberg (1981) [8] described the Prairie Houses designed by Frank Lloyd Wright, while Woodbury and Burrow (2006) [20] represented the Queen Anne Houses using Shape Grammar. While all three research projects developed an accurate grammatical representation to describe specific architectural designs, they didn't introduce Grammatical Evolution as means to explore the design space and evaluate the architectural performance.

Besides the development of generative design strategies [16] in architectural design during the last decades, a research trajectory exploring the application of evolutionary computational systems in architecture [1,4,6,7] emerged. Genetic encoding of design spaces for particular design cases and strategic exploration of representations present critical components of the related discourse. However, a flexible representation through the use of genetic programming widens the scope of the discourse through the expansion of possible design solutions in particular design spaces.

While Koning and Eizenberg [8] are using a parametric shape grammar to describe the Prairie Houses designed by Frank Lloyd Wright and explore the design space generated through this representation, Lee, Herawan and Noraziah [11] are using a similar representation and extend this approach to an evolutionary framework for product design. The shared approach of using a parametric shape grammar is based on the analysis of the design and extraction of main vocabulary elements, before developing rule schemata. Both researchers are presenting the resulting shapes and compositions in the respective papers and describe their modular setup in great detail.

Lee, Herawan and Noraziah [11] are exploring the application of both genetic algorithm and programming in their framework for grammatical evolution in the context of product development. The parametric representation of designs introduces relationships between the elements of the grammar and allows for the representation of complex product designs. The research presented in this paper, applies a similar methodology to the evolutionary design of architectural shapes. Already, Koning and Eizenberg [8] highlighted the potential for application of shape grammars in architectural design and provide insights on how the design space for Prairie Houses could be extended.

The application of graph grammars, as described by McDermott in 2013 [12], is another option to extend the presented shape design process to another representation that is of interest in the context of architecture. Structural systems are of major importance in the development of architecture and structural frames can be represented as graphs. In his research, McDermott [12] points out that graph grammars reduce complexities faced throughout the implementation of shape grammars. Another interesting aspect of McDermotts work in 2013 [12] is the use of graph grammars with multiple rules, consisting of a selector (for node selection) and an action part (modifying the subgraph). This representation adds more flexibility to the generated design outcomes and allows to encode more complex generative processes. Through the use of graph grammars, arbitrary forms usually encountered with direct representations was reduced.

Both, Lee, Herawan and Noraziah [11] and McDermott [12] are integrating artificial fitness evaluation along with automatic evaluation of numerical values derived from the representation. This approach can be used in further research to allow designers direct impact on the search process. An application case for this method would be the interactive exploration of design spaces for mass-customization in architectural design [2,3,19].

## 7 Conclusions

In this study we presented a genetic programming based grammar evolution method for automated shape generation. Three shape generation experiments of two types are presented which are architectural envelope design and facade design respectively. Through this study we have shown the applicability of this evolutionary method to facilitate early stage of architectural design. Implications of this intelligent approach are discussed in this paper. We conclude that automated shape generation by grammatical evolution offers a huge potential for the development of performance-based creative systems.

*Context and site specifity.* At this point of the experiments, the representation of the architectural shapes is independent from context and environment, which would usually impact on the architectural design process. This might also drive the development of a set of fitness functions based on cost, area, volume or other properties of the geometry and the representation.

## 8 Future Research

During the research process, the presented methodology for the development of representations for design processes and the respective implementation of grammatical evolution will be used as basis for the development of an intelligent framework for interactive decision support in architectural design. In this next steps of the research project, the development of an intelligent design framework for interactive decision support during architectural design processes is approached through an abstract reflection on the process to explore the theoretic potential of the process to provide methodologies for architectural design, before strategically extending the implementation of the developed methodology. It will address the opportunities and limitations presented in the results section of this paper.

Another aspect for the extension of the process is the introduction of multi-criteria evaluation as part of the fitness evaluation to incorporate a variety of fitness measures that are meaningful in the context of architectural design, e.g. structural evaluation, evaluation of environmental parameters, introduction of site restrictions and measures that reflect on the performance of the functional program. Another area of interest in the context of fitness evaluation is the evaluation of the representation itself and especially in the context of more complex design outcomes this approach to fitness evaluation will be of high value to ensure the quality of the representation used during grammatical evolution.

The introduction of new strategies to integrate human interaction into grammatical evolution will be taken forward from there through the development of other methods for artificial selection during the evolutionary process.

## References

1. Byrne, J.: Approaches to evolutionary architectural design exploration using grammatical evolution. University College Dublin (2012)
2. Ceccato, C.; Simondetti, A.; Burry, M.C.: Mass-customization in design using evolutionary and parametric methods. In: Proceedings of the 2000 ACADIA Conference (2000)
3. Duarte, J.P.: Towards the mass customization of housing: the grammar of Siza's houses at Malagueira. Environ. Plan. B: Plan. Des. **32**(3), 347–380 (2005)
4. Frazer, J.: An Evolutionary Architecture. Architectural Association, London (1995)
5. Heisserman, J.; Woodbury, R.: Generating languages of solid models. In: SMA 1993 Proceedings on the Second ACM Symposium on Solid Modeling and Applications, pp. 103–112 (1993)
6. Janssen, P.; Kaushik, V.: Evolving lego. Exploring the impact of alternative encodings on the performance of evolutionary algorithms. In: Rethinking Comprehensive Design: Speculative Counterculture, Proceedings of the 19th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2014, pp. 523–532 (2014)
7. Janssen, P.: A design method and computational architecture for generating and evolving building designs. The Hong Kong Polytechnic University (2004)

8. Koning, H., Eizenberg, J.: The language of the prairie. Frank Lloyd Wright's prairie houses. Environ. Plan. B: Plan. Des. **8**(3), 295–323 (1981)
9. Koza, J.R.: Genetic programming. a paradigm for genetically breeding populations of computer programs to solve problems. Stanford University (1990)
10. Langdon, W.B.: Genetic Programming and Data Structures. Genetic Programming + Data Structures = Automatic Programming!. Genetic Programming. Springer, Boston (1998). doi:10.1007/978-1-4615-5731-9
11. Lee, H.C., Herawan, T., Noraziah, A.: Evolutionary grammars based design framework for product innovation. Procedia Technol. **1**, 132–136 (2012). doi:10.1016/j.protcy.2012.02.026
12. McDermott, J.: Graph grammars for evolutionary 3D design. Genet. Program Evolvable Mach. **14**(3), 369–393 (2013). doi:10.1007/s1071001391900
13. Montana, D.J.: Strongly typed genetic programming. Evol. Comput. **3**(2), 199–230 (1995). doi:10.1162/evco.1995.3.2.199
14. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Trans. Evol. Comput. **5**(4), 349–358 (2001). doi:10.1109/4235.942529
15. Poli, R., Langdon, W.B., McPhee, N.F., Koza, J.R.: A Field Guide to Genetic Programming. Lulu Press, Raleigh (2008). lulu.com
16. Roudavski, A.: Towards morphogenesis in architecture. Int. J. Architect. Comput. **7**(3), 345–374 (2009). doi:10.1260/147807709789621266
17. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical evolution: evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–96. Springer, Heidelberg (1998). doi:10.1007/BFb0055930
18. Stiny, G., Mitchell, W.J.: The palladian grammar. Environ. Plan. B: Plan. Des. **5**(1), 5–18 (1978)
19. Williams, N., et al.: FabPod: designing with temporal flexibility & relationships to mass-customisation. Autom. Constr. **51**, 124–131 (2015)
20. Woodbury, R.F., Burrow, A.L.: Whither design space? Artif. Intell. Eng. Des. Anal. Manufact. **20**, 63–82 (2006)