

## 368: Sustainable master planning using design grammars

Jan Halatsch <sup>1\*</sup>, Antje Kunze <sup>1</sup>, Gerhard Schmitt <sup>1</sup>

*Chair for Information Architecture, Swiss Federal Institute of Technology Zurich, Switzerland<sup>1</sup>  
halatsch@arch.ethz.ch*

### Abstract

Successful sustainable urban planning has to deal with existing demands in urban design connected to a highly complex system in every direction of business a planner could think of. Furthermore, reality frequently overtakes ongoing planning. These circumstances are especially severe for the urban planning of mega cities, where sustainability must be in the focus of interest for several reasons. "Green lungs", as one example supporting sustainability within urban environments enhance the functionality of cities to balance humidity locally, to shade buildings and sites and to enable public recreation as well as to reduce air pollution directly and indirectly. This quantitatively and qualitatively less explored branch of urban design requires complex and time consuming modeling tasks even on a master planning scope. In this paper we describe a novel method to create sustainable urban vegetation designs automatically as a kind of early simulation step, which can be used as a guideline for following master planning tasks of high-density urban environments. We had formalized sustainability criteria available in current planning knowledge into CGA shape grammar, which had been introduced by [1]. Additionally, we extended these urban planning rule sets with defined urban landscape patterns. We use these patterns with a procedurally model rule-based distribution and placement of vegetation as well as landscape objects to generate sustainable urban environments. We also show how to use a shape grammar in combination with procedural methods that iteratively develop an urban design, automatically creating more and more details in order to plan sustainable cities more effectively. We link the use of patterns [2] and the design possibilities of shape grammars to plan sustainable design. As results, we present the simulated master plans of different sustainable urban environments in different climates. This paper evolves out of a recent paper [3] of the authors with a strong similarity in section 1-5. Exception: The contributed work is extended by the use of design grammars suited for sustainable design. The extensions adhere especially to the application of design patterns in the context of sustainable master planning and are focused but not limited to vegetation scenarios.

Keywords: sustainable master planning, design grammars, vegetation scenarios

### 1. Introduction

Since the traditional occupational field of architects spread into the domain of digital content creation, their creative capabilities and expert knowledge are increasingly in demand by the entertainment industry. Besides this fact today's challenging master planning projects require architects to merge digital content creation and former analogue design methods. These are time-consuming tasks, which are especially difficult to handle in large-scale landscape and urban planning projects with high demands on the two key features: reliability and quality of plans. It can take several people years to model detailed 3D land- and cityscapes. Next generation applications that are already used in other industries can supplement and partially replace conventional empiric planning methods in architecture and landscape design. They could also dramatically raise design quality through (1) the added predictability of how the sketched landscape design will look like, (2) the added possibility to simulate critical measures for city planning like enhanced airflow through skillfully placed vegetation, and (3) the opportunity to redesign drafted models within seconds what before could have required months of work.

Furthermore, precise design attributes can be integrated in order to achieve sustainable designs.

There are plenty of context-sensitive design decisions and simple repetitive digital content creation tasks. They usually depend on well-defined planning constraints. These tasks should be automated where applicable, since there is no creative expert knowledge necessary to process them. We propose the use of a shape grammar in combination with procedural methods, which is adapted to the needs of architects, landscape and urban planners. In this aspect, this paper introduces a novel approach for a grammar-based planning of open spaces in urban environments as an inevitable part of a sustainable city creation framework.

#### 1.1 Related Work

A landmark in the formal theory of architecture was the introduction of shape grammars by Stiny [4]. These shape grammars can be used to analyze and describe designs for a wide range of architecture, such as Palladian villas [5], Mughul gardens [6], Frank Lloyd Wright's prairie houses [7], or Alvaro Siza's houses at Malagueira [8]. However, the applicability of Stiny's original

shape grammars as a practical modeling tool is unclear, because they are hardly amenable to computer implementation. Mayall [9] presented an interesting implementation of a shape grammar specified for landscape design. The system produces urban scenes by generating parcels with simple 3D buildings and by distributing vegetation objects. Müller et al. [1] presented a more recent approach. They introduce a novel attributed shape grammar called *CGA Shape*, which is suited for applications in computer graphics. This framework allows the encoding of a wide range of architectural designs and the automatic detailed generation of 3D models (Fig. 1).

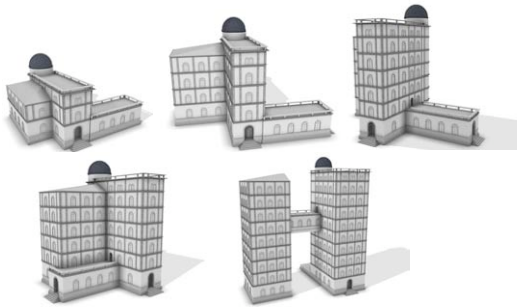


Fig 1. Application of CGA shape, a shape grammar for the procedural modeling of computer graphics architecture. The grammar generates procedural variations of the building mass model (in this case an observatory by Gottfried Semper in Zürich, Switzerland) using volumetric shapes and then proceeds to create facade detail consistent with the mass model.

Shape grammar rules that create certain architectural configurations can be grouped into collections within libraries. Such libraries exist in literary form and are described as *patterns*. The use of patterns in architecture is founded on Alexander et al. [2]. For example, Crowe and Mitchell [10] practically applied Alexander's patterns to describe specific landscape designs. The corresponding key components of patterns in landscape design have been specified and illustrated in detail by Bell [11]. Condensed as illustrated patterns, Turner [12] gives a comprehensive overview of landscape attributes in an urban context. Alexander [13] describes the formal relationship between landscape design and urban planning.

## 1.2 Overview

This paper is organized as follows: Section 2 starts with an overview of the CGA Shape grammar, and afterwards we introduce our novel extensions needed for the encoding of urban open spaces. Section 3 deals with the design of vegetation patterns and their implementation by using our grammar. In section 4, we describe two design studies as examples. Finally we give conclusions and future work in Section 5.

## 2. A Shape Grammar for Sustainable Planning

### 2.1 CGA Shape

In the following, we briefly introduce the main concepts of CGA Shape but refer the reader to [1] for a more comprehensive description. The CGA Shape framework consists of the shape definition, the production process, the rule notation with shape operations, and an element repository.

*Shape definition:* A shape consists of geometry, a symbol, and attributes. The most important attributes are the position  $P$ , three orthogonal vectors  $X$ ,  $Y$ , and  $Z$ , describing a local coordinate system, and a size vector  $S$ .

*Production process:* The production process can start with an arbitrary configuration of shapes, called the *initial shapes*, and proceeds as follows: (1) Select an active shape with symbol  $A$  in the set, (2) choose a rule with  $A$  on the left hand side to compute a successor for  $A$  resulting in a new set of shapes  $B$ , (3) mark the shape  $A$  as inactive and add the shapes  $B$  to the configuration and continue with step (1).

*Rules:* The CGA Shape production rules are defined in the following form:

```
id: predecessor: condition
→ successor: prob
```

where *id* is a unique identifier for the rule, *predecessor* is a symbol identifying a shape that is to be replaced with *successor*, and *condition* is an optional guard (logical expression) that has to evaluate to true in order for the rule to be applied. For example, it can be secured via *condition* that the rule is only applied if the *predecessor* shape is not occluded by another shape. To make designs stochastic, the rule is selected with probability *prob* (optional). Several different types of *shape operations* can be applied to modify the successor shape, e.g. transformations like translate or scale. The insert command  $I(objId)$  adds, according to the size of the current shape, an instance of a geometry object with identifier *objId* (from the element repository). The *subdivision split* can subdivide objects along an axis. For example  $Subdiv(y,7,2,5)\{ A | B | A \}$  splits the current scope along the  $Y$ -axis in three parts with sizes 7, 2, and 5 and adds three shapes  $A$ ,  $B$ , and  $A$  accordingly. To enable size-independent rules, we also include the letter  $r$  to denote relative values, e.g.  $Subdiv(y,1r,2,4r)$  creates three parts: the middle part has absolute size 2 and the remaining two parts split the remaining space in a 1:4 proportion. We also employ a *repeat split*, e.g.  $Repeat(x,2)\{ A \}$  places as many  $A$  shapes of approximate size 2 as possible along the  $X$ -axis of the current shape. The *component split* can divide a shape into its geometric components like 2D faces or 1D edges, e.g.  $Comp(e)\{ B \}$  splits the predecessor shape into edge shapes with symbol  $B$ .

*Element repository:* The library of 3D models consists mainly of basic primitives, elementary architectural objects (e.g. ionic capitals) and plant models. The elements are hierarchically organized in categories and types and each element has a unique identifier, shader attributes and metadata e.g. for sustainable design.

## 2.2 CGA Shape Extensions for Sustainable Design

With the CGA Shape framework a big diversity of hierarchical landscape designs can be encoded mainly by using the subdivision, repeat and component split. Nonetheless, for an intuitive application especially in landscape and cityscape design we have to extend the CGA Shape grammar with two novel shape operations.

The first extension deals with the huge variety of different city objects in the element repository (in this case limited to plants). In our vegetation example, each tree type is classified into three different ages (young, medium and adult age). And for each of these categories, different appearances of the trees are stored. As a consequence, we extend the insertion operation `I(objId)`. Instead of accessing the objects only via `objId`, we use `I(category, type, instance)` to place the tree models. For example, `I(birch, adult, rand(7))` selects randomly one of the first 7 instances of adult birches and inserts this object in the 3D model. Figure 2 shows an example of plant models in the element repository.

The second extension deals with the distribution of instances. We experienced several situations where it is not possible to randomly distribute small shapes (e.g. trees) within a bigger shape (e.g. parcel). Hence we introduce a new shape operation `Distr(density, deviation){ v }` which places point shapes `v` in the current shape. If the parameter `deviation` is 0 the points are homogeneously distributed, otherwise the points are Gaussian distributed (both according to the `density` value).



*Fig 2. Example models of myrtles in the element repository. On the left two full-grown shrubs (approximately 5 meters high); in the middle two examples at medium age (2.5 meters high); and on the right two young myrtles (1.5 meters high).*

The use of design patterns comes in addition. The noted extensions firstly can be combined in a semantic manner to describe certain sustainable design configurations in given context, like vegetation, types of buildings or active building facades. These patterns can be applied to different city scales and will be described more precisely in section 3.

## 2.3 The City Engine System

The CGA Shape grammar with extensions is implemented in a tool called CityEngine [14, 1]. It can process urban environments of any size i.e. ranging from a single lot up to a whole city. The input data is represented in a GIS format and consists of different regions like streets, parcels, building footprints, patios etc. Furthermore, these regions contain metadata information specified by the user in the GUI of the CityEngine. Depending on the metadata attributes, the regions trigger the selection and application of corresponding shape grammar rules. Therefore, the region itself (usually a polygon) is fed as initial shape into the grammar engine, which derives it into an elaborate design by applying the selected rules. The resulting model can then be previewed in the OpenGL viewer of the CityEngine or photo-realistically visualized in a 3D application like Autodesk's Maya.

## 3. Patterns

### 3.1 Enhancing traditional design cycles with shape grammars

One way to classify the term 'design process' in design disciplines is the definition of a design problem and it's solving. Therefore the designer evolves the problem within a cyclical process into a solution, which has many facets and is bound to the designer's intuition. This cyclical process often starts with a general understanding of a complex issue and is usually accompanied by the (re-)definition of its sub-issues. Within these iterative cycles the designer develops a better understanding of the problem's key components and their interrelationships. The results are solutions based on rational, logical and also intuitive insights. The implementation of these solutions involves the development and the realization of the underlying design idea, and its integration into physical and cultural contexts. The sustainable design of urban ecosystems comprises especially the making of places [15]. If we see places as a certain set of designs, which can be decomposed into subordinated problems and their associated solutions, then we can take into account that some of the subordinated problems may be reoccurring problems with reoccurring general solutions. Subordinated solutions are grouped as sets within major solutions, which are in our example spatial arrangements of plant objects in an urban context. In this view, major solutions contain important sustainable design features, which can result in spatial groupings, the alignment and the distribution of the vegetation species and the references to their geometric objects as examples for sustainable configurations. Alexander et al. [2] describes such sets of objects and features as design patterns. Thomas [16], Weysen and De Baere [15] present frameworks and measures for sustainable design patterns, which serve as an inspirational input for our approach. Our vegetation grammar is capable to interpret patterns as described above. Furthermore patterns can also have hierarchical links between them and stay thus human

readable. Since a designer is more or less aware of practical use of patterns in his design, he should easily be able to integrate an automated pattern driven generation approach.

### 3.2 Patterns for Sustainable Planning

Sustainable planning in urban environments operates on distinct major patterns. In order to give landscape planners access to a manageable view in our vegetation approach, we have simplified these patterns into the following regions: avenue zones, block zones, lot zones, patio zones, and also vegetation regions (Fig. 3).



Fig 3. Top: Aerial shots of a typical urban environment.  
Bottom: Typical urban scene with different regions  
(orange: avenue zone, green: block zone, dark green:  
patio zones, blue: lot zones).

Because of the modular implementation of the underlying system, these regions can be set up and, or further decomposed for other aspects in planning sustainable cities.

Avenue zones are straightforwardly created along streets. They span over several blocks if necessary and can be split by crossing streets. They are vegetation-only zones within the generation process, so they do not take part in the creation of buildings nor interact with their footprints. Block zones are vegetation zones enclosed by streets. They have an identical shape as the CE block polygon, but do not include buildings or building footprints, since they are drawn of the pipeline before the creation of housing structures. Block zones are supposed to be containers for parks or small urban forests. Lot zones are required when a combination of buildings and vegetation is desired. They are allocated to the regions' list after the generation of buildings and thus hold information about their footprints, providing means to interact with its housing environment. Vegetation lot zones additionally get building type attributes from the City Engine framework. Attributes like commercial, industrial and residential can be

used later for a further selection of suitable sub-patterns.

A lot zone without a street edge is characterized as a patio zone. It serves to fill the empty space in the center of blocks, enclosed by buildings. Although patio zones share the possibility of lot zones to include footprint information, their primary purpose is to contain vegetation without housing. Areas that span several blocks or cover big parts of the city and contain elements of vegetation are classified as vegetation regions. Examples of these regions include forests within city areas, big parks or fields.

### 3.3 Arbitrary Linkage of Patterns

In our example, patterns consist of sets of attributes, which contain all necessary textual information to describe urban open spaces. From a system's view the description of each pattern – the grammar rules – can be stored within a corresponding pattern text file. Each pattern text file can reference well defined to subordinated patterns. This permits the easy exchange of certain features like climate zone, detail level of the achieved models, for example to provide high resolution polygon models for visualization needs, or low resolution primitives for the use in simulation purposes like airflow analysis.

Since the description of patterns starts at a high level major pattern, which contains a more human readable description, and ends at a low level grammar description, our system offers the possibilities to keep things simple for the non-grammar-expert and to keep things open for users that want to delve into the features of the grammar.

## 4. Examples

### 4.1 Garden Of Versailles

The garden of Versailles, finished around 1700 by order of the monarch Louis XIV is considered the prime example of French Baroque Style in landscape architecture. Its repeating, geometric appearance proves to be an ideal object for a description using rules and patterns of our grammar.

The rules set of the "Versailles-pattern" for this example starts with an occlusion rule, which assures that the open space of the center area does not get filled with items of the following productions, e.g. trees of the crossing avenues. Next, the region is subsequently split into sub-regions, each containing a center avenue and left and right grass zones. In the Versailles example, three recursive split steps subdivide the major region into eight sub-regions (Fig. 4). Once a sub-region reaches the maximal recursion count, the general pattern `grassArea` sets the final style by applying a grass ground cover and distributing hedge elements at the edges. By passing the recursion depth to the `treeAvenue` pattern, avenue parameters like street width, tree density and tree type are adjusted automatically to the importance of the avenue.

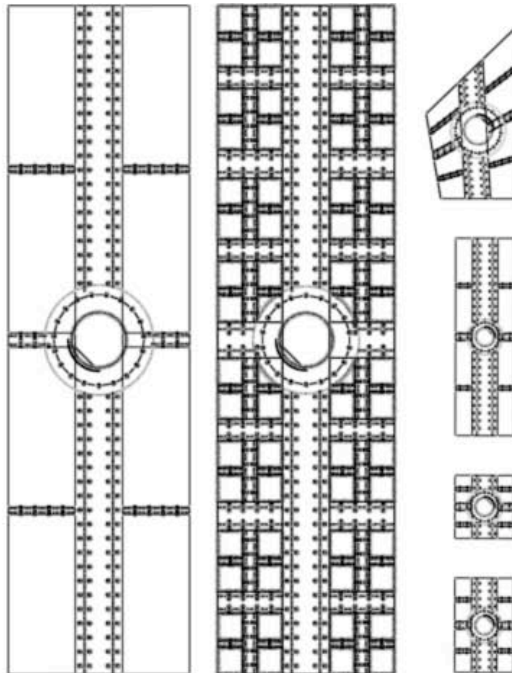


Fig 4. The gardens of Versailles. Top: rendered view of automatically generated reconstruction. Down: original layout of the plan, modified layout with 6 recursive splits, original layout placed on different regions.

#### 4.2 Suburbia: Beverly Hills

Our second example shows a suburban environment, inspired by Beverly Hills. Different garden patterns in combination with stochastic rules ensure diversity in detail views, whereas major pattern descriptions provide the typical homogeneous appearance of the area when observed from a bigger distance (Fig. 5).

Context-sensitive grammar rules allow more precise details in lot and garden patterns: Regions are aware of bordering streets thanks to edge attributes and distribute sidewalks and avenue trees accordingly; footprints of buildings are accessible in vegetation rules and assure correct dimensioning and placing of surrounding elements like forecourts, fences and trees; special markers on lot and building (entrance, garagefront, pool) allow logical connections between house and lot features, e.g. doorways from street to garage.

Due to the modular pattern-based description of lot types, forecourt forms, border styles and fence

appearances and a stochastic mixing of these elements and its controlling parameters an almost infinite visual variety of building lots is generated.



Fig 5. Views of Suburbia in different scales (pictures on left show a complete suburban vegetation solution, on right show the diversity of automatically generated examples).

#### 5. Conclusion and future work

Automatic and semi-automatic approaches can drastically reduced and ease complex and time consuming design and modeling tasks in the planning of sustainable urban designs. Therefore we presented a stochastic shape grammar for Landscape Architecture as a novel approach for the automatic creation of sustainable scenarios in real or virtual 3D cities with the use of vegetation patterns as a given example.

We showed that landscape architects could integrate this approach into their existing design pipelines quite easily for the planning of elaborated greenway and open space systems. The rule-based distribution and placement of vegetation and landscape objects in urban environments enables the planner to apply further simulation steps and is also open for iterative (re-) design purposes. Successful design findings or of suitable configurations can be stored within pattern collections. Also the complexity of procedural modeling became manageable with the help of reusable patterns, which are not limited to landscapes and landscape objects. We adopted our grammar to a promising shape grammar standard the CGA shape grammar by Müller et al. [1]. Our approach can be used for pre-visualization, master planning, guided design variation and for digital content creation purposes of the entertainment industries. Ongoing projects deal with the pattern-based procedural design of the Science City ETH's open spaces in Zurich, Switzerland, and also with the challenging planning of the Singapore-ETH Centre.

#### 6. Acknowledgements

We would like to thank Andreas Ulmer and Pascal Müller for their continuing support and for many helpful discussions as well as the review team.

## 7. References

1. Müller, P., P. Wonka, S. Haegler, A. Ulmer and L. Van Gool, (2006). Procedural Modeling of Buildings. In *Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics (TOG)*, ACM Press, Vol. 25, No. 3: p. 614-623.
2. Alexander, C., S. Ishi Kawa and M. Silverstein, (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.
3. Ulmer, A., J. Halatsch, A. Kunze, P. Müller, L. Van Gool, (2007). Procedural Design of Urban Open Spaces. In *Proceedings of eCAADe 2007*: p. 351-358.
4. Stiny, G., (1975). *Pictorial and Formal Aspects of Shape and Shape Grammars*. Birkhauser Verlag, Basel.
5. Stiny, G. and W. J. Mitchell, (1978). Counting Palladian plans. *Environmental and Planning B*, 7: p. 189-198.
6. Stiny, G. and W. J. Mitchell, (1980). The Grammar of Paradise: on the generation of Mughul Gardens. *Environmental and Planning B*, 7: p. 209-226.
7. Koning, H., and J. Eizenberg, (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B*, 8: p. 295-323.
8. Duarte, J., (2002). *Malagueira Grammar – towards a tool for customizing Alvaro Siza's mass houses at Malagueira*. PhD thesis, MIT School of Architecture and Planning.
9. Mayall, K. and G. B. Endhall, (2005). Landscape Grammar 1: Spatial grammar theory and landscape planning. *Environmental and Planning B*, 32: p. 895-920.
10. Crowe, S., M. Mitchell, (1988). *The pattern of landscape*. Packard, Chichester.
11. Bell, S., (1999). *Landscape: pattern, perception, and process*. E & FN Spon, London.
12. Turner, T., (1996). *City as landscape. A post-postmodern view of design and planning*. E & FN Spon, London.
13. Alexander, C., (2002). *The nature of order: An essay on the art of building and the nature of the universe*. Center for Environmental Structure, Berkeley, CA.
14. Parish, Y. I. H., and P. Müller, (2001). Procedural modeling of cities. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, E. Fiume, Ed.: p. 301-308.
15. Weysen, E., and L. De Baere, (2005). *A sustainable design approach for the urban ecosystem*. Master Thesis. Departement of architecture, ETH, Zurich.
16. Thomas, D., (2002). *Architecture and the urban environment*. Architectural Press, Oxford, UK.