GRAMMARS OF DESIGNS AND GRAMMARS FOR DESIGNING

Grammar-based patterns for urban design

JOSÉ N. BEIRÃO,
Faculty of Architecture, TU Lisbon, Faculty of Architecture, TU Delft

JOSÉ P. DUARTE,
Faculty of Architecture, TU Lisbon

and

RUDI STOUFFS
Faculty of Architecture, TU Delft

Abstract. Analytical work has demonstrated the potential of shape grammars for capturing rules embedded in existing design styles, and generating designs within such styles that match given design contexts. However, the creation of grammars for new design styles, from exploratory rules to design synthesis, remains elusive. The combined use of patterns and discursive grammars is here proposed as a way of encoding the semantics behind recurrent urban design operations and enable the development of a tool to support the creation of new grammars. The idea is that by exploring the combination of generative patterns designers may arrive at new grammars.

1. Introduction

The research described in this paper is part of a larger project called City Induction. The overall goal of this project is to create a system that is able to generate urban solutions for specific design contexts and it encompasses three subgoals: (1) to read data from the site context on a GIS platform and then formulate an adequate program for the urban intervention; (2) to generate alternative design solutions that match the given program, and (3) to evaluate evolving design solutions against the program to obtain satisfactory results. As such, the project integrates three modules, namely, the urban program formulation module (Montenegro and Duarte, 2008), the design generation module (Beirão et al, 2008) and the evaluation module (Gil and Duarte, 2008).

This paper is focused on the generation module and, particularly, on its main operational concept, the urban patterns. It briefly describes the urban ontology common to the three modules and then defines the common structure of urban patterns. The latter is illustrated with a specific case for clarification. The urban ontology will be further detailed in a future paper,

The grammar-based generation module is foreseen as a tool for supporting urban design. Shape grammars (Stiny and Gips, 1972) have been used essentially for analytical purposes as a means to understand the rules underlying given design styles (e.g., Stiny and Mitchell, 1978a, Koning H, Eizenberg J, 1981). Although foreseen in Stiny and Gips' seminal paper, only in some cases they were used as an exploratory tool in the design process (Stiny, 1980; Knight, 1999). Even in fewer cases were they used for such purpose in the urban design process (Beirão and Duarte, 2005). Departing from a clear distinction between 'grammars of designs,' and 'grammars for designing' the first being analytical grammars and the second 'original grammars', this work proposes a methodology for using grammars for generating original design solutions. The idea is not the exploration of design solutions within a given grammar, but the progressive development of a new grammar for a new design context. Considering that this is how designers operate in current design processes, a tool for supporting such processes, should take this distinction into account.

To achieve such a goal, it is not sufficient to explore an existing grammar in a linear and deterministic way, as usually occurs in analytical works. Although, a grammar enables diversity and flexibility by exploring its design rules, determinism needs to be avoided to allow the designer to find his personal expression as the result of an exploratory process that leads to design synthesis. In fact, the designer might transport a grammar from design to design, transforming and refining it in each design process, and a new grammar might emerge during the design process. In addition, the design tool is intended to be used by different designers with different design languages and so not to be bound to a specific language is crucial.

On the other hand, in urban design it is possible to identify recurrent patterns (Alexander et al., 1997) which can be encoded into a computational structure (Gamma et al., 1995). We show that each design pattern might be encoded into a discursive shape grammar (Duarte, 2005)..Therefore, we propose a system that encodes into discursive grammars urban patterns commonly used by designers, thereby creating an abstract compound grammar for urban design. This abstract grammar can be used for developing specific urban grammars by making particular arrangements of urban patterns and instantiating them during the design process.

Section 2 describes the concept of urban patterns. Section 3 identifies the precedents of the current work. Section 4 briefly describes the urban ontology on which the CityInduction system is based and Section 5 explains how urban patterns are encoded into discursive grammars. Section 6 illustrates with a specific case study how urban patterns grammars can be used for designing. Finally, Section 7 lists the conclusions of the current work and identifies the following step.

2. Urban patterns

As described by Schon (1983), the designer develops his ideas following the interaction between what he sees and what he draws, while sketching during the design process. He uses stimuli taken from the context and the underlying constraints to define his design strategies. Also, well known cultural characteristics about the context, either from experience or tacit knowledge, suggest the designer to put forward design assumptions known to solve problems. In doing so, the designer is in fact recognizing urban patterns and defining urban patterns to use in the design.

The use of the concept of patterns has been quite successful in computer science since Alexander's book 'A Pattern Language' was published in 1977. The concept of identifying recurrent problems and providing them with typified flexible solutions has been captured and reused, particularly in object oriented programming. The 'design patterns' concept as developed by GoF (Gang of Four – Gamma et al, 1995) has played an important role while defining a rigorous pattern structure suitable for computer programming that added accuracy to Alexander's patterns. Both concepts of patterns are important to our work. Alexander's approach tells us how to build up a language for designing, creating new patterns and understanding recurrent cultural characteristics that can be reused to design. The GoF gives us the structure and accuracy required to use patterns in programming and developing the link between the general descriptions of patterns and the operations required to materialize them.

3. Urban grammars

Shape grammars and patterns have been previously used in urban design as a teaching methodology in two design studios (Beirão and Duarte, 2007) that took place at the Faculty of Architecture, TU Lisbon in 2002-2003 and 2003-2004. A detailed survey of this study can be found in Beirão (2005). In these studios, students' procedures and workflow as well as results revealed interesting common characteristics. The following aspects were observed: (1) each team elected a specific pattern as the main conceptual pattern involved in the design generation to define the implicit development vision and constrain the application of other patterns; (2) each team elected a different set of patterns based on their reaction to the design context to define their own design language; and (3) the teams organized the rules into thematic sets which were conceptually related to the selected patterns. It was also observed that teams used different rules to express their vision of the same pattern, which means that a pattern can have different formal approaches, and thus lead to the use of different rule sets. This is the creative potential we want to explore.

4. Urban Ontology

4.1. ONTOLOGICAL STRUCTURE

An urban ontology organizes and defines the significant relations among the various types of objects and features found in urban space. It was developed after the analysis of several urban plans used as case studies (Beirão et al, 2008) and it is used as a basis for developing the next steps in this work. So far, the ontology has the structure shown in Figure 1. Future extensions are foreseen to integrate the research results of the formulation and evaluation models.

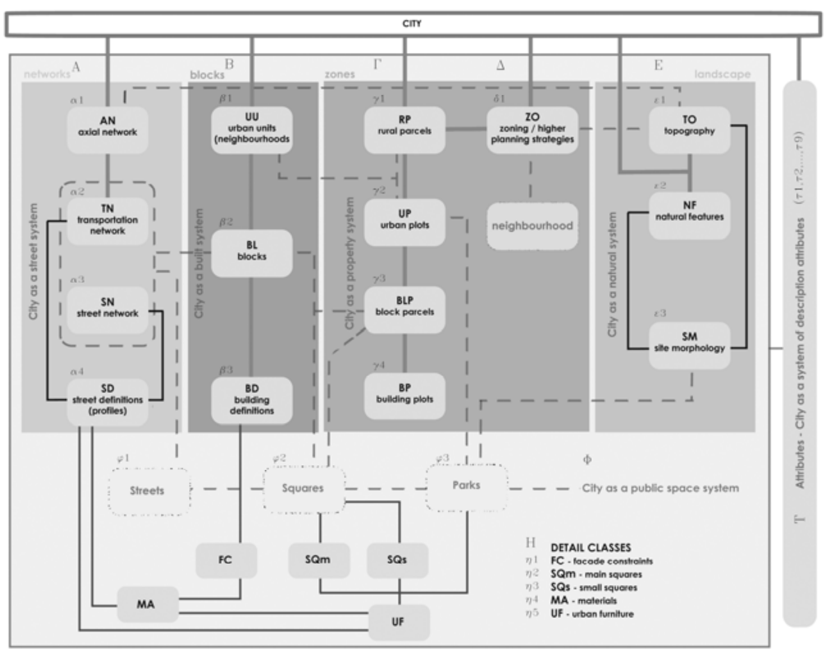


Figure 1. An ontology for urban design.

The urban ontology is divided into seven sub-ontologies for objects and one sub-ontology for attributes each one embedding features from a specific context of the city structure. ‘City as a street system’, for instance, is a sub-ontology for the city networks and it is denoted with the Greek capital letter Α. Each sub-ontology or system has a shape representation (Α’) and a shape description (α). Every system has a set of object classes, each class has object types and each object has a set of parameters and attributes. Classes are denoted with two bold capitals. Lines in Figure 1 indicate the existence of relationships between object-types from different classes. The systems form parts of the ontology as branches or interlaced branches.

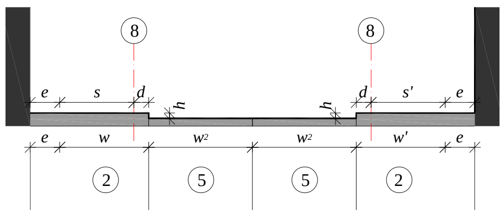
Sub-ontology Α, for instance, describes the street system. It is composed of four object classes, the axial network, **AN**, the transportation network, **TN**, the street network, **SN**, and street definitions, **SD**, which refer to their sections as compositions of profile components. The **AN** provides the axial description of the city. It should not be confused with the axial map from space syntax studies, as it just defines the compositional structure setting four axis types hierarchically structured from a_1 to a_4. TABLE 1 shows a subset of the sub-ontology Α, namely **TN** and **SD** object classes. The left column shows the object classes while the right column shows some of the object types within each class. Every street type in **TN** is defined as a collection of street profile components from the **SD** class. Each street type is composed of a minimum set of profile components which constitute its minimum requirements. Each street type has also a set of additional profile components that can be added to these minimum requirements until it fulfils the maximum width allowed for the street. The minimum requirements for a Local Distribution street type **S2** are (see the street profile components list in TABLE 1 and the schematic section in Figure 2):

S2: ② | 2 x ⑤ | ② [① ③ ④ ⑤ ⑧].

Features between square brackets correspond to additional components that can be added to the obligatory structure of this street type. There are many possible combinations but they cannot exceed the maximum value set for this street width.

TABLE 1. Detail of the Street Network and Street Definitions classes showing some of the object types within each class – the table has been simplified.

Groups of entities – Object Classes	**Entities** (components of urban space) – Object types
α2 – **TN** - Transportation network	**Street types:** **R1** – <u>Ring Roads</u> / **R2** – <u>Structural Street</u> / **S1** – <u>Distribution</u> / **S2** – <u>Local Distribution</u> / **S3** – <u>Local Access</u>
α4 – **SD** - Street definitions (street profile components)	**Sp - Street profiles:** ① - street parking / ② - sidewalks / ③ - bicycle lanes / ④ – bus lanes / ⑤ - car lanes / ⑧ - tree alignments



$1.25 \leq w \leq 5.0$; $s \geq 1.2$

$w = s +$ ⑧$width + d$

$0.3 \leq d \leq 0.75$

e is an extra space for additional purposes: e.g. – esplanade, benches, etc.

$0 \leq e \leq 2.5$

w and *w'* may be different

for w_2 see TABLE 1

Figure 2. Schematic section for S2 street type – minimum profile indicating the variables for the profile components.

The relationships between components are dependent on qualitative criteria and are defined through basic standards taken from the regulations and quality data bases. The qualitative relationship criteria will define minimum, recommended and optimum standards for each component and absolute maxima in some situations. The data is taken from Pedro (2002) because this information is already organized according to the three mentioned standards. The street types can be symmetrical or asymmetrical arrangements of the street profile components that respect specific standards and allowed relationships between components. For instance, it is not possible to place a street parking profile in between two sidewalks.

A detailed description of the sub-ontologies using the sorts formalism (Stouffs et al, 2007) will be the subject of forthcoming papers.

4.2. ONTOLOGICAL OBJECTS

Objects have a specific meaning that can be understood by the urban designer. They represent meaningful parts of the city. The objects are used by the generation module as shapes manipulated by shape grammars. The definition of a common structure for objects allows for a clear relationship among the elements in the ontology and for the generation module to recognize them, their meaning and, therefore, their role in the shape rules. An object will be defined by a 3-tuple with the following format: (*Geometry*, *Marker*, *CoordinateSystem*). The *Geometry* is a parametrical representation of a city feature and has specific parameters. The *Marker* places the object in the ontological context and defines its meaning and behaviour through attributes. The *CoordinateSystem* links both to a specific coordinate system and enables one to link the object with its representation on a GIS platform.

There are two types of attributes: quantitative and qualitative. *Quantitative_Attributes* assign data values to objects (e.g., *Volume*, *NumberOfFloors*, *ConstructionArea*, *BuildingAge*, etc.). These values are directly linked with the geometry. *Qualitative_Attributes* assign a qualitative value to the urban entities (e.g., *Function*, *HistoricalBuilding*, *Landmark*, etc). These attributes are properties of the objects which can be used to influence the generation process. *Initial_Attributes* is a subset of *Qualitative_Attributes* and contains the attributes that can be used to trigger the initial patterns in the generation process.

A *Marker* contains both types of attributes plus an *IDList* of the object which is an ID code identifying its position in the ontology. A *Marker* is also a 3-tuple with a specific format: (*IDList*, *Quantitative_Attributes*, *Qualitative_Attributes*). The ID code is a list of strings including 4 components: Object type, Object number, Class name and Sublevels, which is a list indicating the allowed sublevels or object classes for establishing future operations. The complete ID for **S2** is: (**S2**, **Nr**, **TN**, (**SN**, **SD**, **BL**, **SQm**, **SQs**, **UP**, **BLP**)). Note that links with higher levels of the ontology are always admitted, although not indicated in the ID code. For each object there is one single *Marker.*

5. Codifying urban patterns into discursive grammars

The basic concept for City Induction is an extension of the discursive grammar formalism, adapted to fit urban design (Duarte, 2005). Urban designs are obtained by applying an urban grammar which is a specific arrangement of urban patterns. An urban pattern generates designs using a short discursive grammar operating with objects that belong to the ontology. As a discursive grammar it has a shape grammar part (Stiny and Gips, 1972) and a description grammar part (Stiny, 1980). The design generation involves the application of rules codifying typical urban design operations into a shape grammar in such a way as to match requirements specified in urban programs formulated using a description grammar. An urban grammar is therefore a compound grammar composed of several discursive grammars, each corresponding to an urban pattern.

Formally, an urban pattern is an adaption of Alexander et al (1977) and Gamma et al (1995) concepts. Gamma et al propose that patterns should be composed of 13 parts: *Name* / *Intent* / *Also Known As* / *Motivation* / *Applicability* / *Structure* / *Participants* / *Collaborations* / *Consequences* / *Implementation* / *Sample Code* / *Known Uses* / *Related Patterns*. We leave the detailed explanation of each part to the authors' definition and point only our proposed adaptations.

An urban pattern is composed of 11 parts: *Name* / *Intent* / *Also Known As* / *Known Uses* / *Motivation* / *Applicability* / *Structure* / *Participants* / *Collaborations* / *Discursive Grammar* / *Related Patterns*.

Name / *Intent* / *Also Known As* are equal to Gamma's definition. *Known Uses* appears in this position because it corresponds, to a certain extent, to Alexander's archetypal illustration. We only add an explanatory caption to the illustration. *Motivation* / *Applicability* follow the same principles as in

Gamma. *Structure* includes a graphic representation of the classes used in the pattern and an algorithm to generate variations of the pattern. *Participants* define the initial shapes and which objects are used in the pattern, their relationships and parameters. This is defined through the *IDList* of the object. *Collaborations* correspond to *Quantitative_Attributes* and *Qualitative_Attributes* and they assign behavioural criteria to the objects. So, *Participants* / *Collaborations* are the collection of participant objects and active attributes, which are used to define the matching function in the shape rules. Gamma's *Consequences* / *Implementation* / *Sample Code* are replaced by a *Discursive Grammar*. The discursive grammar encodes the shape descriptions, shape rules, and parameters used to generate the pattern. It also has additional heuristics to search for the best rule application in each iteration. *Related Patterns* follows the same logic as in Alexander and Gamma, but in this case we can define *Related Patterns* as those patterns that use as initial shapes objects belonging to an instantiated pattern. This is defined in the sublevels in the ID code.

6. Designing with urban patterns grammars

As mentioned in the introduction, designers combine patterns in their processes. Patterns correspond, in fact, to recurrent operations used by designers and which they can refer to using their names or an iconic pictogram. The structure we propose for urban grammars permits one to establish the relation between what designers perceive and the formal representation of the shape rules involved in the urban pattern.

Briefly, in our system, designers select patterns from a pool of available urban patterns to produce their designs. They create their own language by selecting patterns and by imposing constraints on the variable parameters associated with the grammar rules defining the patterns. Language is therefore, dependent on two factors: the specific selection of patterns and their parameter specifications. To provide this behaviour, patterns are defined in the most abstract possible way, so that they can be as generic as possible, aiming at a wide application leaving personal interpretations for the designer to define by imposing constraints on the associated parameters. Also, the attributes related to initial patterns must have a very wide application.

The initial patterns in the design derivation should encompass recognizable features in the territory to support the first rule application. This means that the initial shape in a design and within the initial pattern has to be universally recognizable in order to be applied. The problem is that the shape is never the same because all sites are different. What is common to any intervention site is that they all have a boundary which we will call the Intervention Site Limit (labeled I_s) and elements that the designer selects as references to support the design. These elements will be labeled with R_{ef} to denote that they constitute an attribute *ReferenceFeature*. Viewpoints, roads, monuments, buildings, and trees can constitute *ReferenceFeatures* as long as they are selected as such by the designer. These features can be linear or focal and thus be represented by polylines, lines or points. The automatic recognition of these features is not pursued at this stage as we assume that the designer will select them at the start of the generation process by labeling them. Note that I_s is always a closed polyline, new designs will always be applied inside this limit and the initial attributes R_{ef} may be applied to features inside or outside I_s.

Following these principles, all patterns that are amenable for initiating the generation process have rules applying to features containing *ReferenceFeature* attributes and therefore, their application may be considered universal. Each step in the generation process corresponds to a pattern application. After each step, the system shows what patterns may be applied next. The selection of patterns to apply might be automated if descriptions of the next design goals are provided, but this process will not be addressed in this paper. In any case, all patterns must have clearly defined end states and initial states compatible with the end states of other patterns. Note that attributes are properties of the ontological objects and labels assign attributes to shapes. Labels are key to trigger shape rules. A shape may have several attributes but each attribute is represented by only one label.

The following example illustrates the above concept showing a complete urban pattern. For a better understanding of the pattern and to contextualize its application, Figure 3 shows the case study used to extract urban pattern grammar rules. It is an urban plan designed by the Portuguese architect Chuva Gomes for an extension of the city of Praia in Cape Verde. In the rules, shapes with dashed lines correspond to shapes extracted from the case study, thereby allowing one to recognize its application in the plan. These shapes always have the same attributes, but may have different configurations. They are in fact labels that start the generation process. The parametric shape part relationship in the rules is applicable independently of the specific shape taken from the case study. Finally, the names of the patterns were chosen from common, widely accepted, urban concepts so that they are not perceived as particular to this specific case study but usable in other cases as well.

The Praia design starts by applying a common and very old urban pattern – the *Cardus* and *Decumanus* – two orthogonal axes oriented towards the cardinal points, namely the *Cardus* indicating the north south direction. The *Cardus* is also the longer line in the composition guiding axes. Chuva says he was looking for a long line able to structure the whole composition. He sets the *Cardus* as an extension of an existing road and the *Decumanus*, by defining a perpendicular to *Cardus* passing through a reference point, the top of a hill, that stands just outside the Intervention Site (I_s) as a visually relevant landmark. We split the pattern in two, *Cardus* + *Decumanus*. The patterns description is not complete due to space restriction, but the omitted parts are not essential to understand their concept and formal description.

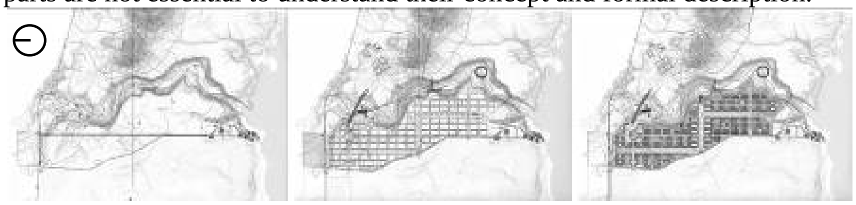


Figure 3. Chuva's design sequence: a- Pre-existences (axes were added); b- Creating the grid; c- Squares, building definitions and details. Drawings by Chuva.

Pattern Name / (001, Cardus)

Intent / Sets the main compositional axis a_1 within I_s.

Known Uses / It was used in Roman cities as the main compositional axis. It is a recurrent feature used all along the history of urbanism.

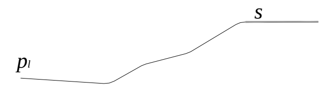
Applicability / The pattern is used to start a design. It sets out the first and main compositional axis, an a_1 axis type, within I_s. When the pattern is selected and triggered, it prompts the user to select features in the territory to be used as reference for the pattern application. The designer's selections are labeled references, R_{ef}. The composition is defined through the first few axes: a_1 and a_2 axes. So, the selected references will not be erased until all the a_1 and a_2 axes are generated – the pattern *cardus* does not erase the reference attributes.

Participants / Uses as initial shapes pre-existing objects from any class marked as R_{ef} and the intervention site limit (I_s). The pattern creates **AN** objects – type a_1. $\alpha_1 : \varnothing \rightarrow a_1$

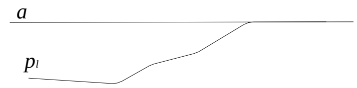
Shape Grammar / Rule 01, draws all possible axes *a*, that can have relations with the references. Rules 01a to 01c simply apply three different spatial relations. Other rules may be added to extend the pattern application but these are the ones that better describe Chuva's design process. Rule 02 trims each axis *a* to fit inside the intervention site I_s and changes the label *a* to a_n. Rule 03a selects the set of the longest generated a_n axes and maps them into an a_{sel} axis as defined through the description rule $\alpha_1 : a_n \rightarrow a_{sel}$. All axes that are at least 90% the length of the longest a_n are selected. Rule 03b erases the remaining a_n axes. Rule 04 picks the a_{sel} axis which defines the smallest angle with the north-south orientation and maps a_{sel} into an a_1, *cardus* axis. Rule 05 erases all reminiscent a_{sel} axes.

Rule (number)	Rule (representation)

Rule 001-01a



=>

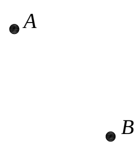


$\forall p_l : p_l \in \{R_{ef}\}$

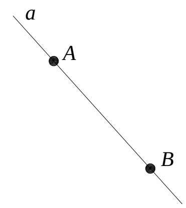
$s \in p_l \wedge s$ is maximal , $|s| > 10\% |p_l|$

$\alpha 1 : \varnothing \to a$, $a \in \mathbf{AN}$

Rule 001-01b



=>

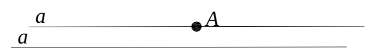


$\forall A, B : A, B \in \{R_{ef}\}$

$\alpha 1 : \varnothing \to a$, $a \in \mathbf{AN}$

Rule 001-01c

=>

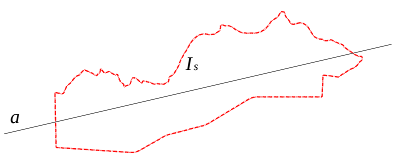


$\forall A : A \in \{R_{ef}\}$

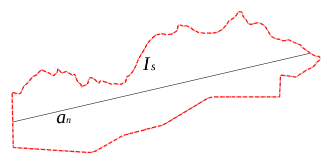
$a \,//\, a$

$\alpha 1 : \varnothing \to a$, $a \in \mathbf{AN}$

Rule 001-02



=>

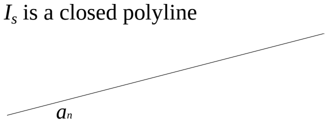


$a, a_n \in \mathbf{AN}$, $n \in \mathbb{N}$

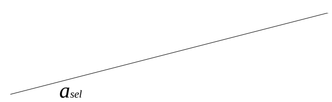
I_s is a closed polyline

$\alpha 1 : a \to a_n$

Rule 001-03a



=>

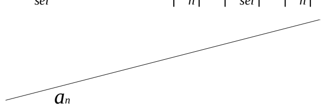


$\forall a_n \in \mathbf{AN}$, $n \in \mathbb{N}$

$\exists a_{sel} \in \mathbf{AN}\textbf{:}\ \mathbf{90\%} |a_n| < |a_{sel}| < |a_n|$

$\alpha 1 : a_n \to a_{sel}$

Rule 001-03b

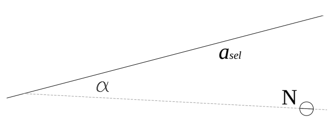


=> Ø

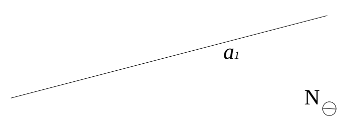
$\forall a_n \in \mathbf{AN}$

$\alpha 1 : a_{sel} \to \varnothing$

Rule 001-04



=>



$\forall a_{sel} \in \mathbf{AN}$

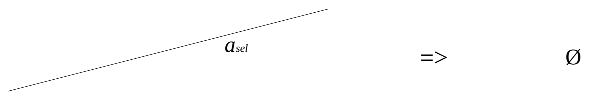
if $\measuredangle \alpha$ is minimum

$\alpha 1 : a_{sel} \to a_1; cardus$

$a_{sel}, a_1, cardus \in \mathbf{AN}$

Rule

001-05



$\forall a_{sel} \in \mathbf{AN}$ $\alpha 1: a_{sel} \rightarrow \varnothing$

Related Patterns / Decumanus

Pattern Name / (002, Decumanus)

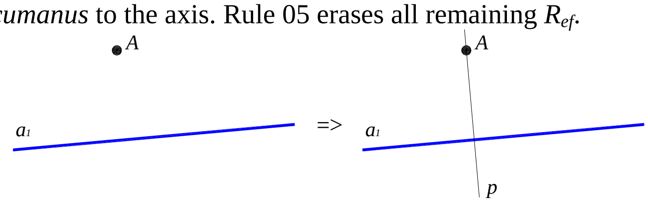
Intent / Sets a perpendicular axis (a_2) to the *Cardus* axis within I_s.

Applicability / The pattern ends the *Cardus* and *Decumanus* composition. It sets out the second compositional axis inside I_s. a_1 and a_2 axis types are considered to be compositional axes. When a *decumanus* pattern is applied there are already references within the drawing otherwise the pattern will not be legal for application. A *decumanus* is an a_2 axis.

Participants / Initial shape is an a_1 axis and a *cardus*. Uses pre-existing objects marked as R_{ef} and I_s. Creates an a_2 axis. $\alpha_1 : a_1 \rightarrow a_1, a_2$

Shape Grammar / Rule 01 draws perpendicular axes (p) to the *cardus* passing through all points labeled R_{ef}. Rule 02 trims all p axes outside I_s. Rule 03 selects from 2 available p axes the one closer to the middle point M of the *cardus*. The rule applies recursively until it selects only the closest to M. It can be argued that there might be other criteria for this rule but other rules may be added to the pattern in order to extend its application. This was simplified in this version of the pattern. Rule 04 assigns an a_2 label and *decumanus* to the axis. Rule 05 erases all remaining R_{ef}.

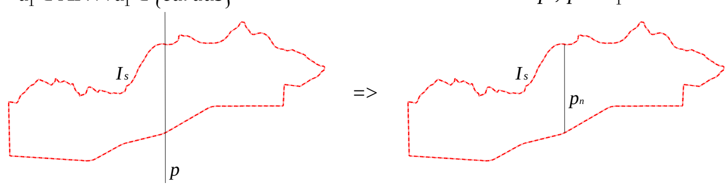
Rule 002-01



$\forall A : A \in \{R_{ef}\}$ $\alpha 1 : a_1$

$a_1 \in \mathbf{AN} \wedge a_1 \in \{cardus\}$ $\alpha 1 : \varnothing \rightarrow p\ ,\ p \perp a_1$

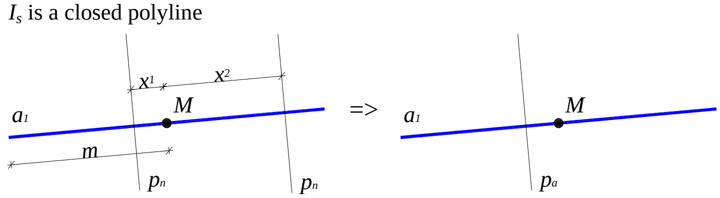
Rule 002-02



$p, p_n \in \mathbf{AN}\ ,\ n \in \mathbb{N}$ $\alpha 1 : p \rightarrow p_n$

I_s is a closed polyline

Rule 002-03



Rule 002-04

$\forall p_n \in \mathbf{AN}, \exists p_n : x_1 < x_2$, $n \in \mathbb{N}$

$a_1 \in \mathbf{AN} \wedge a_1 \in \{cardus\}$, $m = \|a_1\| / 2$

p_a

=>

$\alpha 1 : a_1$

$\alpha 1 : p_n \rightarrow p_a$

a_2

$\alpha 1 : p_a \rightarrow a_2 ; decumanus$

Rule 002-05

$\tau 6 : R_{ef} \rightarrow \varnothing$

Related Patterns / Orthogonal Grid

7. Conclusions

In this paper, we set up the basic structure of the City Induction generation module. We describe two structural concepts of the module: (1) urban ontology, a structure of urban objects and features that defines the basic components of the city structure, and (2) urban patterns, a grammar-based formalism that encodes typical design operations used by urban designers. Urban patterns are discursive grammars and, therefore, they have a shape part and a description part. The shape part corresponds to objects in the ontology, whereas the description part encodes object attributes. The design process evolves by combining available patterns.

The scope of our design system is still limited by the set of available patterns which has been extracted from the case study. It may be argued that the range of available patterns is limited, but that is just a matter of finding and codifying other patterns from other case studies to enlarge the pattern-base system. In the process of extending the pattern base, we need to guarantee that the ontology is respected and that new patterns either are starting patterns or apply after existing patterns. In the former case the end shape of new patterns need to match the initial shape of existing patterns. In the latter case, new patterns need to have as initial shapes the end shapes of existing patterns.

In addition, urban patterns need to be defined in such a way as to guarantee that they can be applied to other contexts than the one associated with specific case studies. This means that the parameters of shapes in the urban pattern grammar rules need to take values from intervals wider than those found in the case study. In the process of identifying and codifying urban patterns we are defining a compound grammar composed of all the urban pattern grammars. As a grammar defines a language of designs, this compound grammar defines a high abstraction level language for urban design.

Specific languages of designs can be defined by designers in the design process first by selecting specific arrangements of urban patterns grammars and then by imposing restrictions on their shapes parameter values. For instance, the pattern for generating urban blocks foresees that these can have customizable width and length, but a designer can impose restrictions on their values, as Chuva did in the design for Praia, which consists of 50 m x 80 m city blocks.

Future work foresees the implementation and testing of urban patterns on a CAD/GIS platform.

Acknowledgements

J.N. Beirão is funded by Fundação para a Ciência e Tecnologia (FCT), Portugal with the grant SFRH/BD/39034/2007. His research is part of City Induction research project which is also supported by FCT and hosted by ICIST at TU Lisbon (PTDC/AUR/64384/2006). We would like to thank J. Gil and N. Montenegro for their inputs and discussion and also J. Pedro for his readings and comments. Gil and Montenegro are responsible for the evaluation model and the formulation model, respectively.

References

Alexander, C Ishikawa, S Silverstein M: 1977, *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York.

Beirão, JN: 2005, *Gramáticas Urbanas (Urban Grammars)*, Master Thesis, ISCTE, Lisbon, available at http://www.bquadrado.com/paginas_web/targets/grammars.html

Beirão, J, Duarte, J and Stouffs, R, 2008, Structuring a Generative Model for Urban Design: Linking GIS to Shape Grammars, *Architecture in Computro - 26th eCAADe Conference Proceedings*, Antwerpen, pp. 929-938.

Beirão, JN and Duarte, JP: 2007, Urban Design with Patterns and Shape Rules, *in* EH Stolk and M Brömmelstroet, (eds), *New Town Simulation - 2nd International Seminar, 2007*, Almere, available at http://www.bquadrado.com/paginas_web/targets/grammars.html (forthcoming)

Duarte, JP: 2005, A Discursive Grammar for Customizing Mass Housing: the case of Siza's houses at Malagueira, *Automation in Construction*, **14** (2): 265-275.

Gamma, E, Helm R, Johnson R and Vlissides J: 1995, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA.

Gil, J and Duarte, JP, 2008, Towards an Urban Design Evaluation Framework, *Architecture in Computro - 26th eCAADe Conference Proceedings*, Antwerpen, pp. 257-264.

Knight, T. (1999), Application in architectural design, and education and practice, *Report for the NSF/MIT Workshop on Shape Computation*, Massachusetts Institute of Technology.

Koning H, Eizenberg J, 1981, "The language of the prairie: Frank Lloyd Wright's prairie houses" *Environment and Planning B* **8**(3) 295-323.

Montenegro, NC and Duarte, JP: 2008, Towards a Computational Description of Urban Patterns, *Architecture in Computro - 26th eCAADe Conference Proceedings*, Antwerpen, pp. 239-248.

Pedro, JB: 2002, *Programa Habitacional, vol.4 – Vizinhança Próxima, (Housing program, vol.4 – Close neighborhood)*, LNEC, Lisbon.

Schon, DA: 1983, *The Reflective Practitioner: How Professionals Think in Action*. New York, Basic Books.

Stiny, G and Gips J: 1972, Shape Grammars and the Generative Specification of Painting and Sculpture, *Information Processing*, **71**: 1460-1465.

Stiny, G: 1980, Introduction to shape and shape grammars, *Environment and Planning B: Planning and Design*, **7**: 343-351.

Stiny, G. (1980). Kindergarten grammars: designing with Froebel's building gifts, *Environment and Planning B: Planning and Design*, **3**: 409-462.

Stiny, G.: 1981, A note on the description of designs, *Environment and Planning B: Planning and Design*, **8**: 257-267.

Stiny, G and Mitchell, WJ: 1978, The Palladian grammar, *Environment and Planning B: Planning and Design*, **5**(1): 5-18.

Stouffs, R, Krishnamurti, R and Park, K: 2007, Sortal structures: supporting representational flexibility for building domain processes, *Computer-Aided Civil and Infrastructure Engineering* **22**(2): 98-116, Special Issue on Product Models (eds. A. Watson and C. Eastman).