

An Urban Grammar for Praia

Towards Generic Shape Grammars for Urban Design

José Beirão¹, José Duarte², Rudi Stouffs³

^{1,2}Faculty of Architecture, TU Lisboa, Portugal,

^{1,3}Faculty of Architecture, TU Delft, Netherlands

¹J.N.Beirao@tudelft.nl, jnb@fa.utl.pt,

²jduarte@fa.utl.pt, ³R.M.F.Stouffs@tudelft.nl

Abstract: *This paper presents a shape grammar for planned urban spaces intending an implementation for generative urban design. This implementation will form part of an urban design support tool defined to formulate, generate and evaluate urban designs. The goal is to formulate urban program descriptions according to context conditions using a description grammar and generate alternative design solutions using a shape grammar. The generation is guided by several evaluation processes performed by an evaluation module. In this paper we are focusing on the definition of the generic shape grammar using an existent urban plan as a case study. The aim is to encode the design moves of the urban designer into generic grammar rules amenable for specific instantiations through the customization of rule parameters.*

Keywords: *Shape grammars; patterns; generative urban design.*

Introduction

This paper presents a grammar for designing urban spaces. The research is integrated in a larger research project called City Induction involving two other researchers and aiming at integrating an urban program formulation model (Montenegro and Duarte, 2008), a design generation model (Beirão et al, 2008) and an evaluation model (Gil and Duarte, 2008) into an 'urban design tool'. The idea is to read data from the site context on a GIS platform, generate program descriptions according to the context conditions, and then generate alternative design solutions guided by evaluation processes to obtain design solutions that satisfy the program. The concept is an extension of the discursive grammar schema developed by Duarte (2005) and adapted to urban design. The generation of urban designs involves

the application of urban design rules codified in a shape grammar (Stiny and Gips, 1972) according to the requirements of the urban program codified in a description grammar (Stiny, 1980). The program formulation module is perceived as a system for developing context dependent planning programs, or in other words, a development vision (Friedman, 1997) that fits the context. The generation module is thought of as an intelligent design support tool that generates typical urban design procedures encoded into a customizable and generic design grammar. The evaluation module is intended as a system that guides the design decisions involved in the other two modules. The focus of this paper is on the generation module.

The goal is to infer and codify shape grammar

rules that can be used for the generation of urban grids focusing specifically on the generation of orthogonal grids. Urban designers perform recurrent design moves or design procedures, which we have encoded into generic parametric grammar rules that may be used to instantiate specific design solutions. The sets of rules defining design procedures were initially called Urban Patterns (Beirao et al, 2009a) and later called Urban Induction Patterns – UIP (Beirao et al, 2009b) to emphasize the distinction from street patterns. Non-orthogonal grids and distortions on orthogonal grids using UIPs are the subject of current research.

In the next section, we explain the theoretical concepts supporting our research, namely, the definition of Urban Induction Patterns. In the third section, we present patterns for generating urban grids. Its rules and the derivation of a grid using them are described and explained. Section four sets some conclusions and discusses how variations on the use of Urban Induction Patterns may produce different formal approaches to design.

The generation module and Urban Induction Patterns

The generation module uses Urban Induction Patterns (UIP) to generate urban design solutions. The UIP concept finds its roots in the pattern language concept (Alexander et al, 1977) and the design pattern concept (Gamma et al, 1995), using shape grammars as a formalism to provide formal rule descriptions for the patterns. An UIP is a discursive grammar (Duarte, 2005) that generates a recurrent design move (Schon, 1983) or design procedure in urban design. It is composed of a programming grammar and a designing grammar. The programming grammar is a description grammar that sets the descriptions needed to fulfil a particular design move in a particular urban design context. The designing grammar is a shape grammar encoding the rules to perform the same design move. A set of heuristics guides the shape generation towards a solution that matches

the descriptions defined by the programming grammar. The descriptions in the programming grammar may be set by the formulation model, by direct input from the designer and additionally by data input on regulations and quality standards stored in a data base. An extended description of UIPs may be found in Beirão et al (2009a).

The generation module is composed of three main parts: an Ontology, an Interpreter and an Interface. The Ontology organizes urban objects in a structure of object classes encoding the singular elements playing a role in the urban context. A schematic illustration of the ontology structure may be found in Beirão et al (2009a). The Interpreter is a grammar interpreter which combines and interprets a finite selection of UIPs. The Interface allows the user (designer) to customize the urban grammar options, meaning, to specify the selection and sequence of UIPs and to constrain the rule parameters within the UIPs. Urban grammars are therefore compound grammars for urban design.

The Interface works also as the communication channel with the formulation and the evaluation model, that is, the interface prompts questions or data requests outwards and receives data inputs inwards. The inputs may come from: validation through the evaluation module or prompted by the designer; descriptions provided by the formulation module or prompted by the designer; validation by input of standard values taken from predefined databases or further constrained by the evaluation module; or direct input from the designer.

The next section shows grammars encoding urban induction patterns inferred from an extension plan for the city of Praia in Cabo Verde - Figure 1. To develop the grammars we have used the information provided by the author of the plan, the architect Chuva Gomes, including drawings and sketches from different stages of the design process, final CAD files, and additional information collected in interviews with the designer. The goal was to identify the UIPs that best translated Chuva's design moves and to identify them using names that corresponded

to recurrent urban design procedures, applicable in similar contexts.

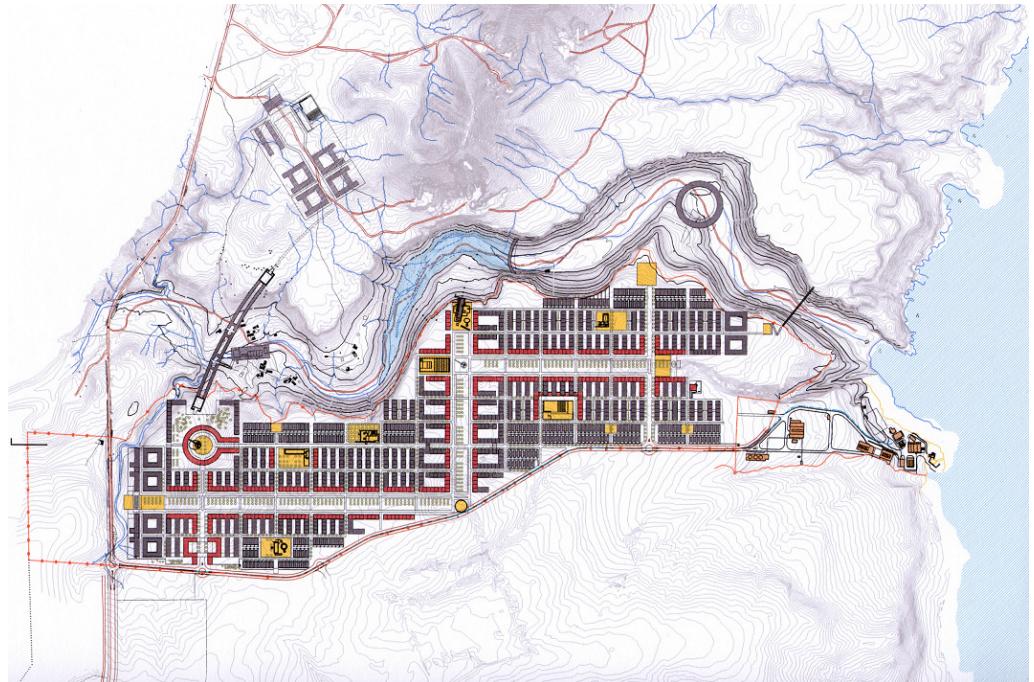
Generating the Praia plan with Urban Induction Patterns

Using the concepts that were explained in the previous section, we started by developing grammars for urban design or, in other words, codifying Urban Induction Patterns as shape grammars. These were developed in an abstract way such that the underlying design procedure was not particular of one specific case but could be used for recurrent design purposes fitting many design contexts, needs, or expressions. We used one case study out of a set of four to extract UIPs. The idea was to define UIPs by encoding the design procedures used in that case study in such a way that variations found in the other case studies could be explained just by changing the selection of UIPs

or by constraining the rule application parameters. We also tried to define them in a way that could be used to explain design cases outside our small set of case studies, thereby increasing their ability to be reused in new design situations. In other words, the case studies were used to bind the universe of possible patterns into a manageable set.

During the analysis of the case studies we concluded that there would be two different ways of producing orthogonal street grids: one by adding axes in one direction and then orthogonal axes (*AddingAxes*), and a second by adding block cells incrementally (*AddingBlockCells*). On a previous paper we showed the generation of the Praia plan using *AddingBlockCells* (Beirao et al, 2009b). Here, we show the generation of the same plan using *AddingAxes*. However, we need to stress that the result is the same in the case of Praia because the architect uses very strict parameters. For instance, the block size is

Figure 1
Extension plan for the City
of Praia (Architect Chuva
Gomes)



always 50m x 80m. Had he allowed the possibility of having different block sizes, which is a possible option within the rule parameters, the results would certainly have fallen into different grid layouts – a subject for future papers.

In order to start the generation process, some initial conditions need to be fulfilled by the first patterns. They have rules operating only with two kinds of shapes: the intervention site limit, labeled I_s and references, labeled R_{er} . The intervention site limit is always a closed polyline in the initial representation and references are elements in the context, inside or outside I_s , which have been selected by the designer as referential elements to support design decisions. As such, I_s and R_{er} are initial attributes of the shapes. The initial UIPs take as the left hand side of their first rule, shapes labeled with I_s and R_{er} attributes. These are the initial shapes of the generation process.

The Praia plan derivation can be described through the following sequence of Urban Induction Patterns: *Cardus* + (*AxisHierarchy_a1*) → *Decumanus* + *Promenade* → *AddingAxes* (x number of axes) + *AdjustAxis* (each time an axis meets the region of a R_{er} point) → *AddBlocktoCell* (x number of cells) → *AdjustingBlockCells* (x number of cells crossing the boundary) → *AddPlaza* → *GeneratePlaza* → *SquarefromBlockSubtraction* (x 3) → *SquarefromBlockTrim* (x 4) → *SquarefromCornerTrim* (x 6) → *TrimPublicSpaceinBlock* (x 3) → *BuildingHeadingAxis* (x 5) → *AddArches* (x 6).

Due to space constraints, the derivation is shown just until the *AdjustingBlockCells* pattern. *AdjustAxis* is, in fact, a set of optional rules within *AddingAxes* (rules 4c, 4d, 6c and 6d – see Figure 2), which allows the designer to adjust the position of an axis in the case it aligns with a region defined by elements previously selected as referential elements for the design. The first patterns, *Cardus* and *Decumanus* were shown in previous papers (Beirao et al, 2009a, 2009b). *Cardus* is a particular case of an UIP called *MainAxisistheLongerLine* where the generated axis is defined according to the north south direction. *Decumanus* is therefore a particular case of the UIP

OrthogonalAxis. The rules showed in this paper are applied after the *Cardus* and *Decumanus* have been generated and describe the generation of the plan for Praia using the UIP *AddingAxes*.

Rules for adding axes to form a street grid – *AddingAxes* and *AddBlocktoCells*

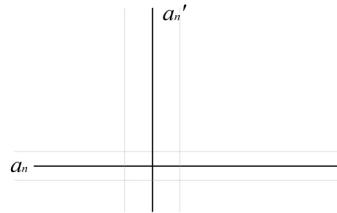
Some rules are omitted because of space constraints if the omission does not affect the full comprehension of the grammar encoding the pattern. Also, some rules are symmetrical to others, namely rules 3b, 4b, 4d, 5b, 6b and 6d are symmetrical to rules 3a, 4a, 4c, 5a, 6a, and 6c respectively. The axes are taken from the Axial Network object class (**AN**) in the ontology (see Beirão et al, 2009a) and they represent a hierarchy of street axes from a_1 to a_4 . The **AN** objects are responsible for the compositional structure of the street grid. The rule application can be better understood following the derivation found in Figure 2 shows the rules of the urban induction pattern *AddingAxes* (in particular, rules 3a, 4a, 4c, 5a, 6a, 6c and 8) and Figure 3 shows *AddBlocktoCells* rules.

Rule 1 of the UIP *AddingAxes* maps a temporary coordinate system, $x0y$, to two existent perpendicular axes a_n and a_n' . In this case, the axes are the ones generated by the patterns *Cardus* and *Decumanus*. Rule 2 extracts from I_s points of maximum and minimum coordinate values considering the new coordinate system. The points are: \max_x = maximum x value in $x0y$; \max_y = maximum y value in $x0y$; \min_x = minimum x value in $x0y$; and \min_y = minimum y value in $x0y$.

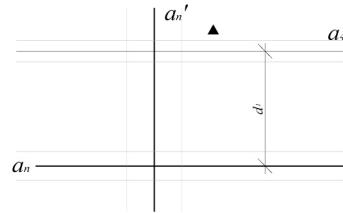
Rule 3a adds a street axis - a_4 - parallel to a_n . The ▲ and ▼ labels are used for the recursive application of rules 4 (a, b, c and/or d) and indicate the direction for applying the rule. Rule 3b is symmetrical to rule 3a and it is applied in the west direction, that is, in the negative y coordinate direction. Rule 4a adds a street axis - a_4 - parallel to an a_4 axis labeled with ▲, erases the label in the original a_4 axis and creates a new ▼ label in the new a_4 axis. The rule applies recursively until it falls outside the intervention site, that is, while $y < \max_y$, where y is the new a_4 y

Figure 2
Urban Induction Pattern
AddingAxes – Rules

Rule 3a



=>



$$a_4, a_n, a_n' \in \mathbf{AN}, n \in \{1, 2, 3\}$$

$$d_1 = w + a_{4_width}/2 + a_{n_width}/2$$

w is the block width

$$|a_4| = |\min x, \max x|$$

Rule 4a



While $y < \max_y$

=>



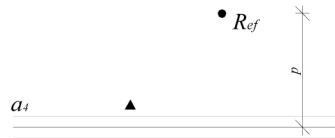
$$a_4 \in \mathbf{AN}$$

$$d_2 = w + a_{4_width}$$

w is the block width

$$|a_4| = |\min x, \max x|$$

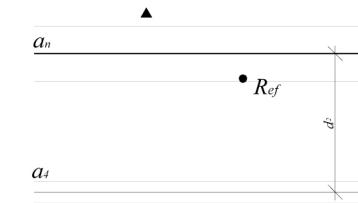
Rule 4c



$$\text{If } \frac{1}{2}w - a_{4_width} < p < \frac{3}{2}w + a_{4_width}$$

While $y < \max_y$

=>



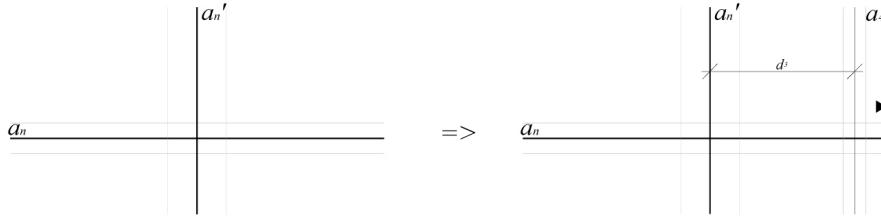
$$a_4, a_n \in \mathbf{AN}, n \in \{1, 2, 3\}$$

$$d_2 = a_{4_width}/2 + w + a_{n_width}/2$$

w is the block width

$$|a_4| = |\min x, \max x|$$

Rule 5a



$$a_4, a_n, a_n' \in \mathbf{AN}, n \in \{1, 2, 3\}$$

$$d_3 = h + a_{4_width} / 2 + a_{n_width} / 2$$

h is the block length

$$|a_4| = |\min y, \max y|$$

Rule 6a



While $x < \max_x$

$$a_4 \in \mathbf{AN}, n \in \{1, 2, 3\}$$

$$d_3 = h + a_{4_width}$$

h is the block length

$$|a_4| = |\min y, \max y|$$

Rule 6c



If $\frac{1}{2}h - a_{4_width} < p < \frac{3}{2}h + a_{4_width}$

While $y < \max_y$

$$a_4, a_n \in \mathbf{AN}, n \in \{1, 2, 3\}$$

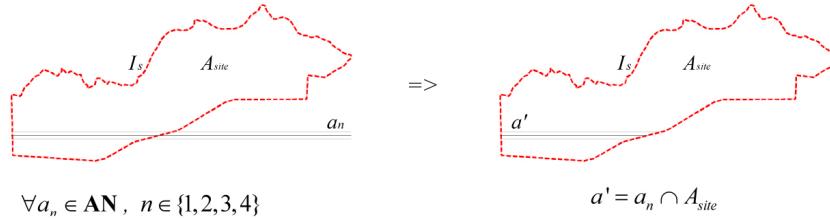
$$d_4 = a_{4_width} / 2 + h + a_{n_width} / 2$$

h is the block length

$$|a_4| = |\min x, \max x|$$

Figure 2 (continued)
Urban Induction Pattern
AddingAxes – Rules

Rule 8



coordinate referred to x0y. Rule 4b is symmetrical to rule 4a and is applied recursively in a similar mode.

In case a selected R_{ef} point or element is in the region of a new axis, the pattern asks the designer whether he does or does not want to apply a new hierarchy to the new axis. Note that the left hand side of rules 4a and 4b rules should actually be the same as rules 4c and 4d. If a R_{ef} point is not found in the conditions stated in rules 4c and 4d then rules 4a and 4b apply, otherwise, 4c and 4d are used. However, this description allows for a better understanding as it isolates the most applied rules, 4a and 4b.

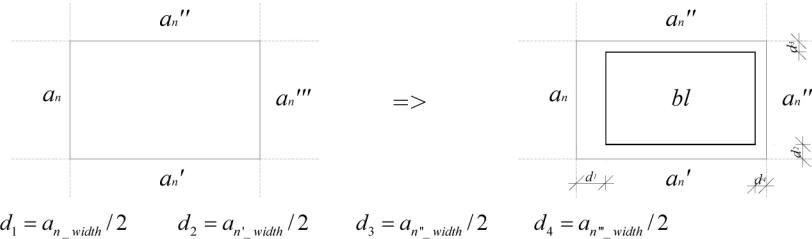
Rule 4c adds a street axis - a_n - parallel to a_4 . The

rule applies recursively, similarly to rule 4a, but following the condition defined for the parameter p . Even if the conditions are fulfilled, rule application is optional, meaning that the designer may choose between applying an a_n axis or the a_4 axis as defined in rule 4a. Rule 4d, which is symmetrical to rule 4c, is omitted.

Rule 5a adds a street axis - a_4 - parallel to a_n' which, in this case, is the one generated by the UIP *Decumanus*. Rule 5b is symmetrical to rule 5a. Rule 6a adds a street axis - a_4 - parallel to the right side of an existent a_4 and it applies recursively within the intervention site limit. Rule 6b is symmetrical to rule

Figure 3
Urban Induction Pattern
AddBlocktoCells – Rules

Rule 1



Rule 2

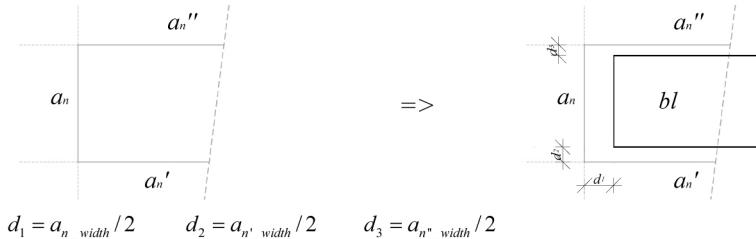
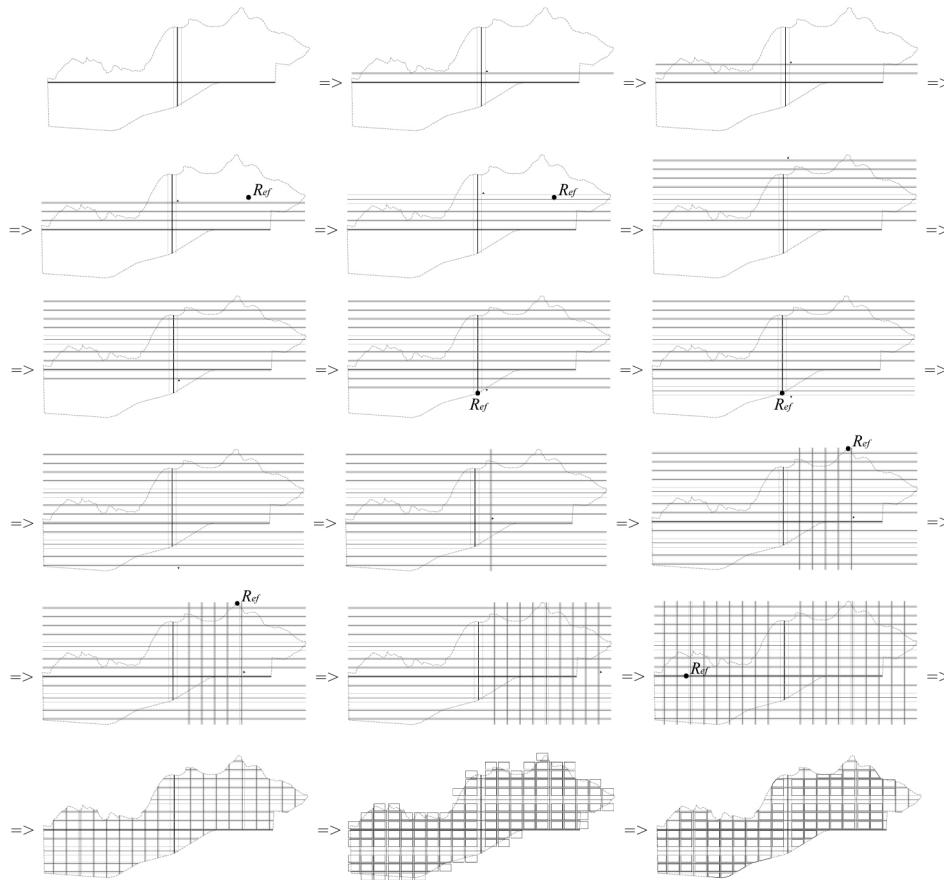


Figure 4
The derivation of the Praia plan



6a. The rule applies recursively until it falls outside the intervention site, that is, while $x > \min_x$ where x is the new $a_i x$ coordinate referred to $x0y$. Rule 6c adds a street axis $-a_n$ - parallel to a_4 if falling in the region of a selected reference element labeled R_{ref} . Rule 6d is symmetrical to rule 6c. Rules 6b and 6d are applied in similar fashion as their symmetrical rules.

Rules 7a, 7b, 7c and 7d erase the remaining \blacktriangle , \blacktriangledown , \blacktriangleright and \blacktriangleleft labels, respectively. Rule 8 trims axes outside the intervention site limit l_s and rule 9

returns to the original coordinate system.

The *AddBlocktoCells* UIP (Figure 3) simply adds blocks to the cells formed by the axes in the grid generated by *AddingAxes*. Rule 1 inserts block parcels in a cell defined by a_n , a_n' , a_n'' and a_n''' street segments where each a_n may be an axis of any hierarchy from a_1 to a_4 . Rule 2 inserts blocks in the limits of the intervention site limit even overlapping l_s . The overlapping blocks may be later adjusted using *AdjustingBlockCells*. The later UIP is shown in Beirão et al (2009b).

Conclusions

The ultimate goal of the City Induction project is a computer platform encompassing modules for formulating, generating and evaluating urban designs. In the present paper we focused on the development of the generation module showing its main generative feature, the Urban Induction Pattern. UIPs are shape grammars encoding recurrent urban design procedures or design moves frequently used by urban designers. They take shapes found in an urban ontology, i.e., a classification of basic elements found in the urban environment, and apply to them transformations that correspond to recurrent urban design moves. A specific selection of UIPs and specific instantiation of their rule parameters defines an urban grammar able to generate urban designs according to a specific urban design language. The underlying idea is to start the design with a very general shape grammar containing a potentially very large set of specific grammars. The concept is that the language may be defined and matured along the design process because the process is simply a continuous request for UIPs and the specifications that the UIPs' grammars need. The synthesis of new grammars is a sequence of the decisions made by the designer along the design process. This permits the use of shape grammars during exploratory design phases. Finally, the general grammar does not cover all the possible design solutions but we believe that it covers a large scope of current design solutions and it may be extended by enlarging the set of available urban induction patterns.

The generation module is foreseen as an autonomous urban design tool using UIPs to generate designs. Its integration with the formulation and the evaluation module has been thought as the bases for a City Information Modeling (CIM) system by analogy with Building Information Modeling (BIM). Such tool is expected to support decision-making on various levels and by different participants involved in urban planning processes.

Acknowledgements

The City Induction project is funded by Fundação para a Ciência e Tecnologia (FCT), Portugal (PTDC/AUR/64384/2006), hosted by ICIST at TU Lisbon, and coordinated by José P. Duarte. J.N. Beirão is funded by grant SFRH/BD/39034/2007, from FCT. Jorge Gil, and Nuno Montenegro are responsible for the evaluation and the formulation modules, respectively. We would like to thank Henco Bekkering, Sevil Sariyildiz and Frank van der Hoeven, for their readings, comments and support.

References

- Alexander, C Ishikawa, S Silverstein M: 1977, *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York.
- 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, J, Duarte, J and Stouffs, R, 2009a, *Grammars of Designs and Grammars for Designing – Grammar based patterns for urban design*, *Proceedings of CAAD Futures'09 Conference*, Montreal, Canada (forthcoming).
- Beirão, J, Duarte, J, Gil, J and Montenegro, N, 2009b, *Monitoring urban design through generative design support tools: a generative grammar for Praia*, *Proceedings of the APDR Congress, 2009, Cidade da Praia, Cabo Verde*, (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.
- Friedman, A: 1997, *Design for Change: Flexible Planning Strategies for the 1990s and Beyond*, *Journal of Urban Design*, 2, 3, 277-295.
- 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.
- Montenegro, NC and Duarte, JP: 2008, Towards a Computational Description of Urban Patterns, Architecture in Computro - 26th eCAADe Conference Proceedings, Antwerpen, pp. 239
- 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: 1981, A note on the description of designs, Environment and Planning B: Planning and Design, 8: 257-267.