Liu, C.-C. et al. (Eds.) (2014). Proceedings of the 22nd International Conference on Computers in Education. Japan: Asia-Pacific Society for Computers in Education

An intelligent tool to assist architecture students in the early stages of design

Eva MILLÁN[a], María-Victoria BELMONTE[*a], Manuela RUIZ-MONTIEL[a], Juan GAVILANES[b] & José-Luis PÉREZ-DE-LA-CRUZ[a]
[a]*ETSI Informática, University of Málaga, Campus de Teatinos, 29080 Málaga (Spain)*
[b]*Escuela de Arquitectura, University of Málaga, Málaga (Spain)*
*mavi@lcc.uma.es

Abstract: In this paper we present BH-ShaDe, an educational software tool that assists architecture students in the process of learning to design single-family dwellings. To this end, BH-ShaDe generates and proposes housing units schemes, which can serve as starting points in student´s exercises and projects. The tool has been designed and implemented based on the ideas of reinforcement learning and shape grammars. The students can select the more promising schemes (among those generated by the tool), and then transform them in complete architectural plans, and finally use them to develop a complete residential project. To evaluate the validity of this approach, BH-ShaDe has been tested with 78 architecture students of the University of Málaga. The results have shown that the starting points were suitable, diverse and useful, and that they have helped students to learn how to design residential projects.

Keywords: architectural education, software tool, learning to design, meta-cognitive abilities

1. Introduction

In any design task it is not easy to start a new project, so architecture students usually struggle when they need to start a new design and face nothing but a blank sheet of paper. To this end, designers often use inspiration sources. The expression "*inspiration source"* describes the conscious use of different resources or even previous designs, as references for the solution to a problem (Eckert et al., 2000). These inspirations sources or starting points can act as triggers for the generation of new ideas, and even accelerate or facilitate the design process, especially in the case of novice designers. For that reason, computer-aided design tools or design assistants can be very useful in these early stages, allowing the exploration of different alternatives and providing feasible starting points.

To this end, we have developed BH-ShaDe (Basic House Shape Design), a software tool that is able to generate raw, schematic proposals (*schemes*) for housing units according to a given guideline (Montaner and Muxi, 2008). To generate these schemes automatically, BH-Shade uses shape grammars (Stiny, 1980), and to do it in an intelligent way, it uses reinforcement learning techniques (Sutton and Barto, 1998). In this way, architecture students can benefit from the existence of many feasible and varied starting points (basic house schemes) for their residential projects, which have been obtained effortlessly. In order to analyze the usefulness of such starting points in the process of learning to design, we have performed an experiment with a group of 78 architecture students of the University of Málaga, with promising results.

In the next section, we will briefly review some related work. The section 3 describes the software tool and its implementation, while Section 4 is devoted to the description of the experiment performed with architecture students. The results of the study are shown and discussed in Section 5. The paper finishes with some concluding remarks in Section 6.

2. Related Work

The computers have been used in Schools of Architecture and introduced in experimental laboratories for design with varied purposes: as a support to imagination and creativity (Elsen and Leclercq, 2008) (Peng and Jones, 2004) (Iordanova, Tidafi and Paoli, 2007); to assist in teaching and learning tasks

(McGill, 2001) (Jain, Kensek and Noble, 1998) or even to introduce virtual reality in architectural education (Redondo et al., 2011).

In relation to design learning, a number of tools have been implemented and tested. For example, SketSha (Elsen and Leclercq, 2008), which is a tool that helps students in the initial stages of collaborative design by supporting free-hand sketches, drawn in real-time in distant locations on a shared workspace. Another example is SUCoD (Peng and Jones, 2004), a web-based virtual city information system that supports 3D urban design. SUCoD allows students to link their CAD skills with knowledge of urban history to carry out urban design proposals. VRSolar (Jain, Kensek and Noble, 1998) and Shaper 2D (McGill, 2001) are examples of more specialized software tools that complement architecture teaching. VRSolar is a web-based teaching tool that helps in teaching topics related to the movement of the sun and its effects on the built environment, while Shaper2D is a design tool that assists students for learning about shape grammars and their uses in the design process. Other works also report innovative uses of computers in learning how to design. (Iordanova, Tidafi and Paoli, 2007) present a study (using case-based-reasoning) performed with architecture students that use a library of referents during their work on the design of a summer theater. Finally, (Redondo et al., 2011) use virtual reality in architecture teaching practices. They merge pictures and virtual models that allow students create interactive photomontages and facilitate the evaluation of the visual impact of their projects.

Our goal is to assist architecture students in the process of learning to design single-family dwellings. Design problem solving is considered a particularly complex task, since these kinds of problems are ill-structured (Simon, 1973), that is, they are characterized by the absence of a unique, well-defined solution. These types of tasks are difficult to teach. Certainly, an educational software tool that generates and proposes different starting points can be quite worthy for design stakeholders in the first stages of the design process. In this way, even though the system does not provide the students with the traditional resources of the tutorial systems (problem evaluation, reading texts,…), they obtain *feedback* on the rules and guidelines in the form of the generated schemes.

This approach is not new: architects frequently use this resource. For example (LeCuyer, 1996) cite Eisenmann’s use of computer-generated forms as starting-points. The sources of inspiration or starting points more commonly used by architects or designers include: i) visual stimuli, like for example sketches (Goldschmidt and Smolkov, 2006), objects of nature (Demirkan and Afacan, 2012), or pictures (Casakin et al., 2000); ii) text as graphs words –for example, the tool Idea Space System (Seger, de Vries and Achten, 2005) which gives designers a word graph that contains architect’s annotations and semantic associations based on them-; or iii) a mixture thereof, like (Malaga, 2000) that compares the generation of ideas in response to visual or textual stimuli and a combination of both.

However, none the above approaches generate the inspiration sources/starting points automatically. Moreover, to our knowledge, there is no computer-aided tool that provides students with automatically generated starting points. Therefore this is a distinctive feature of BH-ShaDe, which not only generates the schemes automatically, but also uses intelligent techniques and shape grammars to provide an unlimited number of schemes adapted to a given housing guideline.

3. The Tool

In this section we describe BH-ShaDe, a software tool that assist architecture students in the process of learning to design single-family dwellings. To this end, BH-Shade generates and proposes housing units schemes that can serve as starting points in students’ exercises and projects. The tool has been designed and implemented based on the ideas of reinforcement learning and shape grammars. By means of reinforcement learning, an agent learns autonomously how to interact with the environment in such a way that the total reward is maximized. Learning occurs through interaction with the environment, by receiving positive or negative rewards after the execution of an action. The agents learn which sequences of actions (policies) yield a good total reward. In our case, reinforcement learning techniques (namely, Q-learning) are applied to “an agent” (a generator of architectural schemes) to obtain an improved version of the agent, which is able to provide good starting points for the student. BH-ShaDe is based on the formalism of shape grammars, so every action of the agent is the application of a rule of the grammar.

Shape grammars have been widely used in design and specifically in architectural research (Cagdas, 1996). A shape grammar is a formal language that represents visual thinking. To this end there is an initial shape (usually called axiom), and a set of design rules or transformations that can be applied to different shapes. Figure 1 represents an example of an axiom, rule, and five successive applications of the rule, starting from the original axiom.

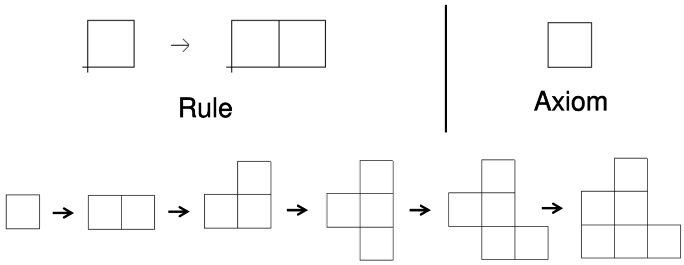

Figure 1. Simple shape grammar that adds squares, with one possible derivation

From the standpoint of generative CAD tools (Chase, 2002), shape grammar-based systems are particularly well suited to easily automate the design, allowing a great deal of exploration.

In what follows we will describe the main characteristics of BH-ShaDe. First we will present the architectural guideline in which it is based. Then we will briefly show BH-ShaDe main features, interface and architecture.

3.1 Criteria for the design of basic houses

Our goal is that students learn to design functionally feasible single-family dwellings. To this end, we have used an existing guideline, originally proposed by Montaner and Muxí architecture studio. This guideline is useful for architecture students to learn how to design dwellings. It also guides the generation of schemes for "*basic houses"* in our software tool. In our context, a basic house is defined as a house that, besides satisfying some minimum habitability conditions, also offers some *adaptability*, i.e., its spatial composition may be modified if the number of inhabitants varies. To generate such basic houses in an *intelligent* way, the tool makes use of reinforcement learning techniques. Therefore, BH-ShaDe is able to intelligently generate two-dimensional floor distribution schemes of basic, two-person housing units. All the schemes produced are distributed over one floor and its total area is restricted to 46 m^2 (as recommended in the guideline).

In Montaner & Muxí's proposal, several kinds of spaces are considered: (1) *specialized spaces* (which need specific installations), (2) *non-specialized spaces* (do not need specific installations, and their use is determined by its inhabitants: dining-room, living-room, bedroom) and (3) *complementary spaces* (such as *distribution hall*, that allows circulation between spaces).

This set of criteria has been implemented as a set of *requirements*. By requirements we mean either constraints (for example, the area of each non-specialized space must be bigger than $9m^2$) or goals (the contour must be as compact as possible). The core of the software tool is a very simple set of rules, together with and an interpreter for them. Random sequences of rule applications would lead to schemes that would not satisfy the set of requirements. To avoid this, violations are automatically detected and *punished*. In this way, and by applying reinforcement learning algorithms, random sequences of rules are gradually replaced by "intelligent" ones (that avoid punishment). The process continues until we obtain a version of the rule interpreter (BH-ShaDe) that generates only *good* schemes. All these *good* schemes are created equal and they are not categorized by the tool. Note that the reinforcement learning process is performed off-line, just once, and prior to the use of the system by students. A detailed description of the interpreter and the learning procedure can be found in (Ruiz-Montiel at al. 2013).

3.2 BH-ShaDe Features

BH-ShaDe can work in two different modes: interactive and automatic. In the interactive mode, it allows an interactive execution of three fixed shape grammars, intended to help students learn the concept of shape grammars. The first one is the toy example shown in Figure 1. The second one generates a very simple housing unit. The third grammar (Figure 2b) is applied to the shape in the canvas when the rules are loaded. The rules then add non-specialized spaces to the housing unit. This grammar makes sense when the shape in the canvas is a housing unit generated with the automatic generation module described below.

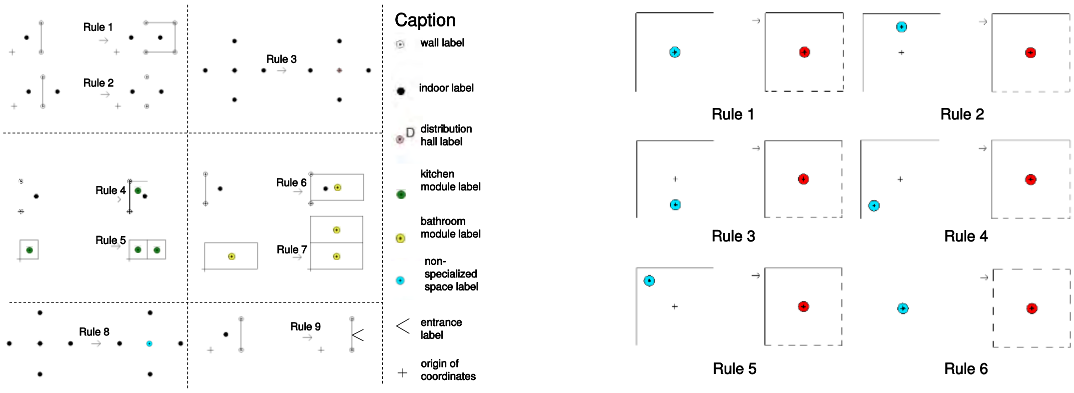

Figure 2. Shape grammars for the generation of a) basic house and b) non-specialized spaces

The automatic mode allows students to obtain housing units schemes according to Montaner and Muxi´s guideline. These schemes are automatically generated by the computer, using the shape grammar shown in Figure 2a (that is hidden to students). In the automatic generation module, students just need to specify the desired number of schemes and then the tool will generate them.

3.3 BH-ShaDe Architecture and Interface

In this subsection we describe the architecture of BH-ShaDe and its interface. In Figure 3 we can see the different modules that interact in order to provide students with the aforementioned features.

BH-ShaDe is built on top of SketchUp (SketchUp, 2013), hence in the bottom of Figure 3 we can see some SketchUp-related modules. Our tool communicates with SketchUp by means of an Application Programming Interface (API). Immediately on top of this API module, we can see a *Shape Grammars Module* that encapsulates the shapes, the rules and the arithmetic for manipulating shapes. On the top, we can find BH-ShaDe interface, that mainly interacts with the two principal modules of BH-ShaDe: *the Interactive Generation Module* and the *Automatic Generation Module*.

The *Interactive Generation Module* deals with the interactive application of shape grammars by means of a general subshape detection algorithm. This algorithm is very powerful as it can detect possible application of rules of any general shape grammar. With the result of this algorithm the Search Algorithm finds a concrete rule application that satisfies the constraints (if present).

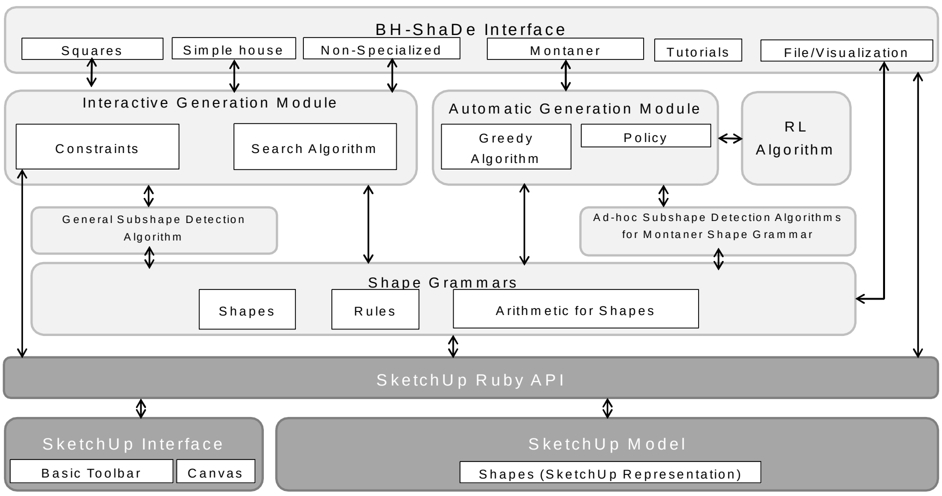

Figure 3. Architecture of BH-ShaDe

The *Automatic Generation Module* applies the shape grammar in Figure 2a by means of an ad-hoc subshape detection algorithm that only works with this shape grammar, but is much faster than the general algorithm. With the result of this algorithm, the Greedy Algorithm chooses the best rule application with the help of the policy generated by the Reinforcement Learning (RL) algorithm. As we explained before, the policies inform about which rule applications lead to solutions that provide the best adjustment to Montaner and Muxi´s guideline. BH-ShaDe interface is built on top of SketchUp interface, as depicted in Figure 4. Shape grammars and the results of successive applications of rules (according to the user's choices or to the automatic generation module) are also shown in the interface.

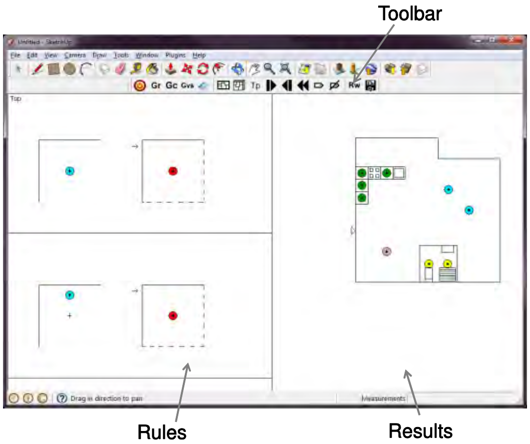

Figure 4. Screenshot of BH-ShaDe interface.

4. The Experiment

In this section we describe the experiment and its settings. The evaluation was performed with three groups of students enrolled in the subject *Architectural Projects VII*, which is taught in the seventh semester of the Architecture Degree of the University of Málaga. In total, 78 students participated in the experiment. Next we will describe the task they had to perform.

First, each student executed the grammar in Figure 2a to automatically generate a basic house. Then, they interactively separated the non-specialized spaces (i.e., placed the walls) using the shape grammar in Figure 2b. The students were then divided in groups of three members. Each group had then 3 schemes, that had been developed by each of their members, partially in automatic mode (the basic house), and partially interactively (walls). This procedure could have continued so students had a large enough collection of basic houses to use in their projects. However, the generation of each scheme consumed some time that was not useful in our overall goal of generating starting points that supported students. Therefore, we decided to use the automatic generation module of BH-ShaDe to automatically generate a further 24 schemes for each group, resulting in a total of 27 schemes per group.

To promote discussion and reflection among the students, these 3-member groups were joined in larger groups of nine students. Each one of these larger groups had then 81 unique schemes that they had to classify according to the following criteria: A (optimal), B (adequate), C (some modifications needed), D (problematic) and E (absurd). Once the schemes had been classified, they had to select the most suitable for different kinds of groupings: row-houses, apartment blocks, galleries, single-family houses. Based on these selections they had to develop a complete architectural project, proposing single-family housing solutions, both in 3D and 2D. The students presented their final projects in the classroom, and the teachers commented and evaluated them.

Finally, the students completed a small survey. The survey was designed as 11 Likert items, relative to four different topics. The students had to evaluate their degree of agreement with each sentence (from 1 -low- to 6 -high-). We have used the more suitable methodology (Jamieson, 2004) to analyze nominal Likert items: mode, median, inter-quartile range and nominal levels of disagree (degree of agreement of 1, 2 and 3) vs. agree (degree of agreement of 4, 5 and 6).

Additionally, the survey included two free-text items where the students could identify the strong and weak points of the software. We have analyzed these free-text items according to the *constant comparative method* (Strauss and Corbin, 1998), a methodology based on *grounded theory*

(Glaser and Strauss, 1967). In the first step, each student's response is decomposed in the ideas it expresses (*answers*). Then these answers are divided into *categories*. In the *phenomenological reduction phase*, the categories are grouped by subject (*themes*). Finally, in the *triangulation phase*, examples of *supporting quotes* are provided. The main advantage of using this methodology is that the ideas expressed in student's answers emerge from the analysis of the sentences, and are not pre-conceived by researchers. Therefore this analysis complements the results obtained in Likert-type surveys.

5. Results

In this section we will present the results obtained in the experiment. They have been organized into different subsections: student's survey, teacher's opinion and student's final projects. Finally, the last subsection is devoted to a global discussion in light of these three sources of evidence.

5.1 Student survey

As aforementioned, 78 students answered the survey. Results are summarized in Table 1, while Table 2 and 3 summarize the results of the two free text items (using the constant comparative method).

Table 1: Results of Likert items.

	1	**2**	**3**	**4**	**5**	**6**	MEDIAN	MODE	(Q_1,Q_3)	Don´t agree	Agree
About the software tool…											
1. I quickly learned how to use the tool	0	3	4	3	**41**	27	5	5	(5,6)	8,97%	91,03%
2. It was easy for me to use the tool	0	0	2	10	32	**34**	5	6	(5,6)	2,56%	97,44%
3. The user interface is intuitive	1	3	12	**30**	20	12	4	4	(4,5)	20,51%	79,49%
4. The tool worked quick enough	1	4	17	**24**	22	10	4	4	(3,5)	28,21%	71,79%
5. The tutorial was easy to follow and useful	1	3	8	11	**36**	19	5	5	(4,5)	15,38%	84,62%
About the schemes proposed by the tool…											
6. The schemes can be helpful in the design process	4	12	17	27	14	4	4	4	(3,4)	42,31%	57,69%
7. The schemes can provide good starting points	2	9	8	25	29	5	4	5	(4,5)	24,36%	75,64%
8. The schemes were interesting	3	7	19	24	21	4	4	4	(3,5)	37,18%	62,82%
About this practice…											
9. All in one, it was interesting	3	5	7	27	**29**	7	4	5	(4,5)	19,23%	80,77%
10. I think that the methodology used was suitable	1	3	15	**26**	25	8	4	4	(4,5)	24,36%	75,64%
11. It was rewarding to work in groups	1	3	12	13	**26**	23	5	5	(4,6)	20,51%	79,49%

Table 2: Categories, themes and supporting quotes for positive aspects.

	CATEGORIES			**THEMES**	**EXAMPLES OF SUPPORTING QUOTES**
	Name	**Nº answers**	**% answers**		
135 different answers	Diversity	26	19,26%	ASPECTS RELATIVE TO QUALITY OF SOLUTIONS (25,18%)	*"Great variety of alternative solutions"* *"What I liked most about this practice is that I could develop a feasible project"* *"The tool provides a great variety of schemes, some of them present little annex spaces that could be grouped generating unexpected solutions"*
	Validity	6	4,44%		
	Versatility for groupings	2	1,48%		
	Usability	3	2,22%	ASPECTS RELATIVE TO THE SOFTWARE TOOL (11,85%)	*"The tool was easy to use"* *"The tool generated the schemes quickly and I could take advantage of some of them"* *"Being able to use new computational methods for architectural design based on randomness"*
	Efficiency	10	7,41%		
	Possibility of using software tools in the design process	3	2,22%		
	Possibility of working in groups	10	7,41%	TEAMWORK (14,82%)	*"Working in groups and new relationships with other students"* *"Debates in the group about what is desirable or not in architectural design"*
	Processes of selection/reflection carried out in the working groups	10	7,41%		
	Randomness	9	6,67%	STARTING POINTS (48,15%)	*"The tool provides a degree of randomness that would be otherwise difficult to include in a project"* *"Little annex spaces that appear accidentally can be used to generate diverse groupings"* *"Being able to have an starting point instead of a blank page"* *"The tool generates schemes that you would not think of"*
	Happy accidents or bugs	3	2,22%		
	Provides starting points	41	30,37%		
	Overcoming preconceived solutions	12	8,89%		

Table 3: Categories, themes and supporting quotes for aspects to be improved.

102 different answers	Software capabilities for edition	9	8,82%	ASPECTS RELATIVE TO THE SOFTWARE TOOL (12,74 %)	*"The user should be able to modify some aspects of the schemes generated"* *"A more intuitive interface"*
	Usability (interface)	4	3,92		
	Shape grammar should include additional criteria	20	19,61%	ASPECTS RELATIVE TO SHAPE GRAMMARS (21,56%)	*"Shape grammars should include additional architectonical criteria"* *"To fully exploit the tool, the user should be able to define his/her own shape grammars"*
	Possibility to include user-defined shape grammars	2	1,96%		
	Overlapping of non-specialized spaces	11	10,78%	ASPECTS RELATIVE TO DISTRIBUTION OF SPACES (17,64%)	*"Overlapping of non-specialized spaces should be avoided"* *"Some kitchen modules are inaccessible"* *"The tool should not generate residual spaces"* *"The location of the entrance door near to the kitchen constraints the variety of the solutions"* *"Better distribution of spaces"* *"Wet zones should be contiguous"*
	Bad distribution of kitchen furniture	3	2,94%		
	Residual spaces	10	9,8%		
	Better location of doors	12	11,76%		
	Better distribution of spaces	3	2,94%		
	Better placement of wet zones	10	9,80%		
	Excess of randomness	9	8,82%	ASPECTS RELATIVE TO THE SOLUTIONS (48,03%)	*"The randomness of the tool should be controlled"* *"Greater variety of schemes generated"* *"Other parameters should be considered (environment, social aspects, etc.) "*
	Poor variety of solutions	7	6,86%		
	Additional criteria should be considered (not only architectonical)	2	1,96%		

As shown in Table 2 and 3, there were 135 different answers for the positive aspects, and 102 for aspects to be improved. From these answers, researchers established a total of 12 and 13 categories, respectively. Two independent researchers assigned quotes to categories. The inter-rater agreement between those two researchers was computed using the *iota* measure (Janson and Olsson, 2001). We obtained *iota* values of 0.707 and 0.898 (a value of 1 represents perfect agreement), which indicates a quite good agreement. Through a negotiation process between the two researches, answers were definitely assigned to categories. The final step of the triangulation process is the *interpretation* of the supporting quotes. In our case, this interpretation will be done in section 5.4, in which we use all the information available to establish some conclusions.

5.2 Teacher's opinion

Three architecture teachers participated in the experiment. One of them was a member of our research team, while the other two did not have previous knowledge of shape grammars or about the tool. We developed a small survey for these two teachers, which had four open questions. For space reasons we do not include their complete answer, but a brief summary of the more relevant information.

It seems that what the teachers liked most of this experience was the possibility to use this kind of tool and learn about shape grammars, together with the interdisciplinary work carried out by the research team and the fact that the tool can provide an unlimited number of schemes, and therefore expedite the design process. Overall, they said that the students had done a great job, and that the more interesting projects emerged from the accidental elements, like the little annex spaces generated in some of the automatic solutions presented by the software tool. More concretely, they said that… *"The tool expedited the design process. The most interesting projects have emerged from accidental elements, like annex spaces or errors. Initially, they seemed not to have any practical use, buy finally they have served to encourage the clustering of the dwellings, and have provided support so the students could freely use their imagination. We do believe the tool has accelerated this kind of discoveries".*

They also pointed out that the use of tool has provided an excellent exercise of analysis and reflection. The students have learned in a practical way that their preconceived ideas are not always the best ones. *"Having 81 housing plan floors automatically generated by the tool, so they could be discussed and selected by the groups of students, has been an excellent exercise about analysis/reflection, which is usually easier to carry out in other people's work than in our own designs. At the same time, the program generates such a wide variety of schemes that accidents occurred randomly, giving birth to what at first sight could be considered as undesirable forms, which finally generated the most interesting projects. Students have therefore learned in a practical way that their preconceived ideas are not always the more appropriate solutions".*

As for possible improvements, they mentioned that it would be useful to include more architectural criteria in the shape grammars defined. *"In relation to the tool and its use in this particular* *activity, we think that it would be desirable to extend the number of variables to be taken into account in the shape grammar".*

5.3 Students final project

In the last session of the experiment, groups of students presented their final projects. Each group presented A1 sheets with the 81 schemes, evaluated from A (no changes needed) to E (the scheme is absurd). In total, there were 9 groups of 81 schemes. The distribution of the percentage of schemes classified according to the different grades (both among those produced by the students and among those generated automatically by BH-ShaDe) is shown in Figure 5.

In their presentations, the students stated that they had held very productive discussions to agree about criteria to classify schemes (recall that teachers also thought that this discussion/reflection process had been very productive). To this respect, some groups had established more demanding criteria than others. The teachers pointed out that some of them had discarded useful schemes for irrelevant reasons, such as a poor positioning of some elements (door, kitchen/bath furniture) or the superposition of non-specialized spaces. The high number of schemes classified with D and E can be explained by the fact that the students considered criteria (circulation, ventilation, light distribution or grouping of wet zones) not accounted in the guideline used by the tool.

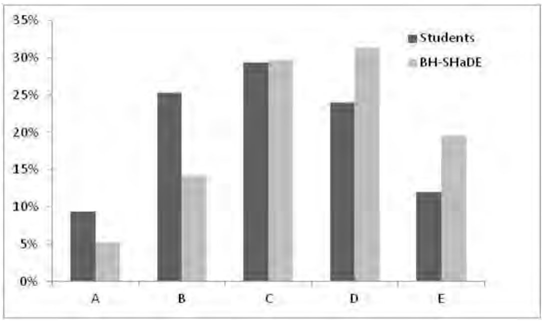

Figure 5. Distribution of the total percentage of schemes of each type (A to E)

From the schemes classified with A or B, each group selected four or five as the basis to create more complex structures. They explored different kinds of groupings: single-family houses, apartment blocks, galleries, etc.

There were many interesting projects, for illustration purposes we will show two of them (selected by the teachers as illustrative for different criteria). The teachers selected the first project as representative of those than emerged from starting points that contained accidental elements (and finally gave birth to creative solutions). This group used one of the schemes generated by the computer classified with A (upper left corner of the Figure 6). The scheme had two small corridors next to non-specialized spaces, and they decided to use these two corridors as terraces to generate an octahedron tower. The second project was chosen because it illustrated a nearly feasible solution (in teacher's words, it seemed nearly pre-conceived). The students selected this scheme (right of the Figure 6 and also rated with A) because it has a good distribution of the so-called wet zones (kitchen and bathroom). In this way, once the schemes are grouped, wet zones occupy the central part of the building.

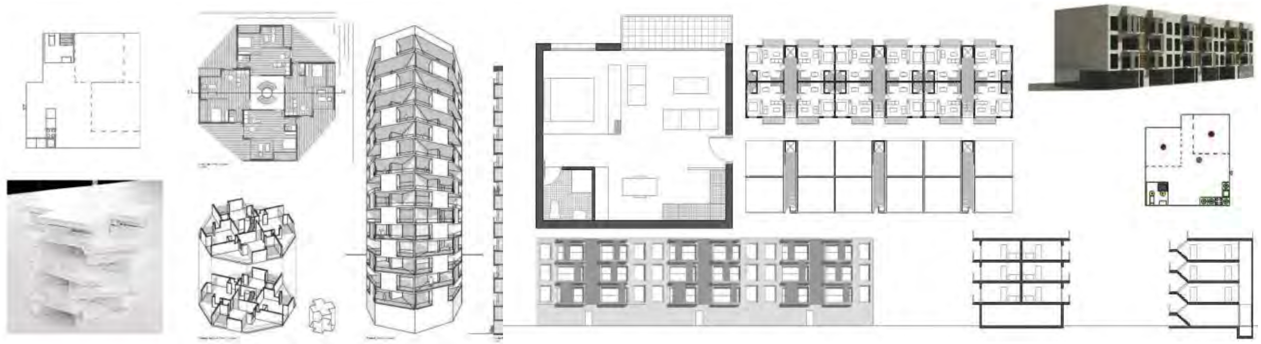

Figure 6. Examples of student's projects: a) Octagonal tower and b) Gallery

5.4 Discussion

In this subsection we will use the results presented in sections 5.1, 5.2 and 5.3 to analyze to what extent the starting points provided by BH-Shade have been useful for the students in the early stages of their design projects.

With respect to this question, probably the more useful Likert items are items number 6, 7 and 8. In particular, item number 7 is very relevant, and we can see that 75,64% of the students agree that "*The schemes provided by the tool can provide good starting points"*. In addition, the students seem to agree that "*The schemes were interesting"* (62,82%), and that "*The schemes can be helpful in the design process"* (57,69%). This same conclusion can be reached from the analysis of the positive aspects of the first free-text item. Indeed, the theme "*Starting Points"* spontaneously emerged from student answers, being mentioned by 48,15% of the students, and specifically 30,37% of them mentioned that "*The tool provides good starting points"*. In this theme, other aspect mentioned by 8,89% of the students was the possibility of overcoming preconceived solutions. Teacher's feedback also seems to support this conclusion, because they said that "*However, the most interesting projects have emerged from accidental elements like annex spaces or errors"*. In fact, and according to their experience, "*the designs of the clusters of schemes obtained using traditional methods are usually more rigid and less creative that the ones generated with the tool*".

Continuing with the positive aspects of the tool, the next more frequently mentioned theme in the survey was "*Aspects relative to the quality of solutions"* (25,28%), and in particular, the category "*Diversity"* (19,26%). As for the teachers, they declared that "*the program generates such high variety of schemes that accidents occurred randomly, giving birth to what at first sight could be considered as undesirable forms, but finally generate the most interesting projects"*. With respect to the tool, the students emphasized (7,41%) its "*Efficiency"* (also the teachers said *"The tool expedited the design"*).

All in one we think that, according to both the teachers and the students, the stronger point of BH-ShaDe is its capability to generate an unlimited number of diverse, feasible, random and suggestive starting points for novice designers.

With respect to aspects to be improved (and focusing our discussion in the quality of the starting points provided), the most frequently mentioned theme was "*Aspects relative to distribution of spaces*" (48,03%), and specifically the categories "*Better location of doors*" (11,76%), "*Overlapping of non-specialized spaces*" (10,78%), or "*Better placement of wet zones*" (9,8%). Next more frequently mentioned theme was "*Aspects relative to Shape grammars*" (21,56%), in particular, the category "*Shape grammars should include additional criteria*" (19,61%). Overall, it seems that both the teachers and the students think that the inclusion of additional architectural criteria (circulation, ventilation, illumination, grouping of wet zones...) could improve the quality of the solutions provided. Finally, and in relation with the overall quality of the starting points, Figure 5 shows that the students gave higher scores to their own designs than to those generated automatically by the tool. This suggests that future versions of the tool should offer a bigger degree of interactivity, allowing the user a greater control in the design of the initial scheme.

6. Conclusions and Future Work

In this paper, we have presented BH-ShaDe, an educational software tool that has been designed to help architectural students in early stages of design, by providing them with starting points for the design of residential projects. The tool is based on intelligent techniques, namely shape grammars and reinforcement learning.

In order to determine the usefulness of the tool we have conducted an experiment in the Architecture School of the University of Málaga. 78 students and three teachers participated in this experiment, which basically consisted in using BH-ShaDe to provide students with interactively and automatically generated schemes to help students to learn how to design dwellings and develop residential projects. Results of the experiment show that both the teachers and the students considered that the schemes provided by the tool were suitable, diverse and useful as starting points, and they had helped the students to develop creative and feasible solutions. Even in some cases, the schemes generated by the tool gave birth to more flexible and innovative projects than those generated with traditional methods.

Regarding future work, the more immediate step is to take into account the results of this study to improve the software tool. Some examples of these possible improvements are: increasing the degree of interactivity of the tool to allow users a greater control of the design process, or including additional architectural criteria in the shape grammar, to improve the quality of the starting points. Finally, we think that the approach exemplified in this work could be used in other design domains as support for the learning process.

Acknowledgements

This work is partially funded by: Grant TIN2009-14179 (Spanish Government, Plan Nacional de I+D+i) and PIE13-052 of the University of Málaga. We would like to thank the reviewers of the paper for their insightful comments.

References

Cagdas, G. (1996). A shape grammar model for designing row-houses. *Design Studies 17*, 35-51

Casakin, H. & Goldschmidt, G. (2000). Reasoning by visual analogy in design problem solving: the role of guidance. Journal of Planning and Design: Environment and Planning B, 27, 105-119.

Chase, S.C. (2002). A model for user interaction in grammar-based design systems. Automation in Construction 11, 161-172.

Demirkan H., & Afacan, Y. (2012). Assessing creativity in design education: Analysis of creativity factors in the first-year design studio. Design Studies 33, 262-278.

Eckert,C. M., Stacey, M. K., & Clarkson, P. J. (2000). Algorithms and inspirations: creative reuse of design experience. Proceedings of the Greenwich 2000 symposium: digital creativity, 1-10.

Elsen, C., & Leclercq, P. (2008). SketSha - the sketch power to support collaborative design cooperative design, visualization, and engineering. LNCS, vol. 5220. Springer Berlin, 20-27.

Glaser, B.G., & Strauss, A. L. (1967). The discovery of Grounded Theory. Chicago IL: Aldine.

Goldschmidt, G., & Smolkov, M. (2006). Variances in the impact of visual stimuli on design problem performance. Design Studies 27, 549-569.

Iordanova, I., Tidafi, T., & Paoli, G. (2007). Is a Digital Model Worth a Thousand Pictures? Computer-Aided Architectural Design Futures (CAADFutures), 85-98.

Jain, A., Kensek, K., & Noble, D. (1998). An interactive web-based teaching tool for simplified 3D analysis of solar rhythms. Automation in Construction 8 (2), 181-194.

Jamieson, S. (2004). Likert Scales: how to (ab)use them. Medical Education, 38 (12), 1212-1218.

Janson, H. & Olsson, U. (2001). A Measure of Agreement for Interval or Nominal Multivariate Observations. Educational and Psycological Measurement 6 (2), 277-289.

LeCuyer, A. (1996). Design on the computer. Architectural Review, January.

Malaga, R.A. (2000). The effect of stimulus modes and associative distance in individual creativity support systems. Decision Support Systems, 29(2), 125-141.

McGill, M.C. (2001). A Visual Approach for Exploring Computational Design (MSc Thesis).

Montaner J.M. & Muxí, Z. (2008). Propuesta de nueva normativa de viviendas. Technical Report, Dirección general de ordenación del territorio, Junta de Andalucía.

Peng, C., & Jones, P.B. (2004). Reconstructing urban contexts online for interactive urban designs. Design Studies 25 (2), 175-192.

Redondo, E., Navarro, I., Sánchez, A., & Fonseca, D. (2011). Visual interfaces and user experience: Augmented reality for architectural education: One study case and work in progress. Vol. 166 of Communications in Computer and Information Science. Springer Berlin, Heidelberg, Ch. 31, pp. 355-367.

Ruiz-Montiel, M., Boned, J., Gavilanes, J., Jiménez, E., Mandow, L., & Pérez-de-la-Cruz, J.L. (2013). Design with shape grammars and reinforcement learning. Advanced Engineering Informatics, Volume 27 2, 230-245

Segers, N.M., de Vries, B., & Achten, H.H. (2005). Do words graphs stimulate design? Design Studies 26, 625-647.

Simon, H.A. (1973). The structure of ill structured problems. Artificial Intelligence 4, 181–201.

SketchUp (2013). http://www.sketchup.com/

Stiny, G. (1980). Introduction to shape and shape grammars. Environment and Planning B: Planning and Design 7(3), 343-351.

Strauss, A., & Corbin, I. (1998). Basics of qualitative research: Grounded Theory procedures and techniques (second edition). Thousand Oaks, CA: sage.

Sutton, R.S., & Barto, A.G. (1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA