Semi-automatic Virtual Reconstruction of Ancient Roman Houses

Telmo Adão, Luís Magalhães and Emanuel Peres
INESC-TEC (formerly INESC Porto) and ECT UTAD
Vila Real, Portugal
telmo.adao@gmail.com, lmagalha@utad.pt, eperes@utad.pt

Abstract— **Video Games may be used with different applications in mind, for entertainment, training and simulation or even as teaching tools. However, the production of contents for such games, namely game scenarios with virtual worlds composed by cities and related elements (streets, buildings, furniture and so on) may require a laborious effort by multidisciplinary teams which lead to an expensive and time consuming solution. In order to avoid the development burden, procedural modeling techniques have been created and applied to generate virtual models of buildings and cities expeditiously with high degree of detail and realism. Nevertheless, most of the existing techniques use complex sets of rules to produce models without being aware of the architectonic structures and its relationships. Thereby, in order to simplify the generation process and rule dealing we propose, in this paper, an ontology-based procedural modeling methodology which integrates an ontological structure, describing city elements and the relationships between each other, and a combined set of procedural techniques to generate roman houses. The preliminary results point to a method capable of producing roman houses expeditiously, including the interiors and the exteriors.**

Procedural modeling system, Procedural modeling architecture, Game Content Generation, virtual representations, Computer-Aided Engineering, Computer-Aided Design (CAD).

I. INTRODUCTION

Video games can be used for learning and training purposes, also known in this context as serious games. This special kind of games is usually used to improve user skills or provide knowledge. In some situations, the realism and the fidelity of the virtual models produced for the serious games are a major concern. The use of serious games to teach History is one of the many examples that can be mentioned to exemplify the demands for faithful realistic models. For history students, this kind of tools may improve learning of ancient history by grabbing their attention and arousing their interest while they explore an interactive virtual world produced to provide knowledge about an ancient place. A serious game may be more appellative than the recommended literature for the subject. However, the production of such contents is still a challenging task which traditionally requires the contribution of multidisciplinary teams composed by several experts and technicians working on the same project for large periods of time. Thus, one can infer that this kind of creative process, as it was presented, is very laborious, expensive and time-consuming. Alternatively, procedural modeling proposes a set of semi-automatic techniques capable of generating virtual environments in a short period of time and with a minimal user interaction.

In the last two decades, procedural modeling has been applied to generate 3D models expeditiously with impressive results. This class of techniques focuses mainly the generation of building exteriors in which the resulting output is a model containing a coherent representation of the buildings, streets and other outdoor elements. However, buildings are more than its exterior appearance. Serious games to teach history or as an archeological tool also require the production of the interiors to provide a more accurate and complete system. In fact, there are a few works that tackle the interiors generation issue by suggesting innovative techniques to achieve this purpose. It is also true that most of these techniques have a lack of support from real rules revealing a disconnection between the models produced and architectonic styles or rules. Concerning archaeological and historical contexts, the referred point is a relevant one because the buildings created across the eras, by distinct civilizations, clearly have differentiated styles which are explicitly related with culture, materials, technology, etc.

Thus, the solution may rely in a combination between an ontological schema, connecting the elements contained in a city with a certain architectonic style, and a proper procedural technique responsible for creating the geometry according to the related architectonic scope. This idea is being followed by us in the development of a system, ERAS [1], capable of generate expeditiously virtual reconstructions of archaeological sites. The ERAS system is composed by two main modules: the information extraction module and the procedural modeling module. In this paper, it will be presented the methodology proposed for the second module and its pilot implementation, which also integrates the ontology suggested in [2]. The proposed methodology joins an L-system with a squarified tree map strategy to generate the building interiors, supported by an ontology. The modeling process begins with the gathering of rules incoming from a simple text file with a proprietary format. Then, an L-System iterates trough the rules in order to recursively split the initial area of the building into smaller ones. This will create a squarified tree map that is the initial outline of the floor plan. The floor plan generation process is over when the rooms are connected to each other by transitions. Next, the walls are extruded and finally the roof is applied. This entire process takes into account the ontology defined, which is used like a supporting data model that holds direct transformations in its geometry. The method was applied to reconstruct ancient roman houses, taking into account the architectural constraints of this kind of buildings.

II. BACKGROUND

Urban Procedural Modeling techniques can be split in two types of approaches: the generation of cities, focusing the exterior of the buildings, and the generation of building interiors. The L-System is probably one of the most known procedural techniques that has been used to produce exterior layout. It was introduced by Lindenmayer [3] and adopted by Parish et al [4] to generate a city. This technique uses an alphabet of symbols combined with a set of production rules. The process starts with an initial set of symbols that are iteratively replaced by other symbols until obtain the final string, which is used to generate shapes using a transformation mechanism. In their work, Parish et. al. used this technique in two steps: one for generating the street network and other for generating the multi shaped buildings. Later, Wonka introduced the split grammar [5] that can be defined as a grammar for tridimensional shaping and geometrically context aware. The process is similar to the L-System but this time the iterative replacement happens directly on shapes until a final state. The process is considered done when all the shapes are final. The process was applied by Wonka and co-workers to detail building façades. CGA shape [6] is another powerful procedural technique that can be used to produce urban virtual environments. The process is still iterative but now, everything starts with a mass model which is the union of volumetric shapes. In a further step it creates building façades properly adapted. In the final step the windows, doors and ornaments are added. The model coherence is a top concerning feature in this technique. These techniques were developed to create urban environments but they can be also used with an archeological purpose in mind. It is the case of the reconstruction of Rome proposed by Dylla and co-workers [7]. The authors used two types of digital reconstruction: Class I for the reconstruction of elements with known position, identification and design, and Class II for general tips about the elements. The procedural technique employed was the CGA Shape. Muller [6] also applied a CGA Shape solution to reconstruct a Puuc-style building according to the ones found in Xkipché, México.

Others tackle the buildings interiors generation problem. Rau Chaplin et al. [8] used a shape grammar to generate floor plans which contain basic rooms. These basic rooms are recognized and arranged in order to define functional zones such as public, private and semiprivate. They use functions to populate the rooms with furniture according to the zone function that matches the predefined boundaries, obtained from a layout library of individual rooms. Martin [9], proposed an approach that uses a graph to represent the division type and the connections between different divisions. The process starts by generating a graph according to the rules defined by the user. Next, each node is mapped to a room. Finally, using a Monte-Carlo algorithm, the division are properly placed and expanded in space. The concept of real time interiors generation was exploited by Hahn and co-workers [10]. The floor plan is generated through a random division of the floor into rectangular divisions and hall passages. The process starts by defining a temporary region which is then divided in smaller temporary and construction regions. This process is repeated interactively until every region becomes construction regions. The architectonic rules guide the process until the generation of the final geometry. Tutenel et. al. [11] used a semantic schema to generate floor plans that are afterwards expanded. Every type of rooms is mapped into classes stored in a semantic library that also define the relationship between rooms, which are basically the adjacencies between them. However, it can be defined other kind of restrictions to force adjacency between rooms such as placing a kitchen right next to a garden. The rooms are placed in a minimum rectangle according to the semantic schema restrictions and then they are expanded until touching the limits of each other. The Squarified Treemaps is a process introduced by Marson and Musse [12] that recursively subdivide rectangles into room divisions. The process starts with a list of values that represents the rectangular areas that should divide the initial rectangular area. Based on it, the proposed methodology splits recursively the starting rectangle into smaller rectangular areas, by picking sequentially the values of the list. The result of the process is a set of rectangular areas representing the room areas. The last step of the floor plan generation is the creation of a corridor which connects the unreachable rooms. Merrel and co-workers [13] developed a method for residential building layout generation, including interiors and exteriors, highly focused on floor plans generation. Their methodology's initial input is a set of high-level requirements, for example the house area and the number of rooms. Then, using Bayesian networks trained with real data, a 2D floor plan is generated. The final step is to extrude the floor plans to get the 3D models, including doors, windows and roof. A constrained growth method for procedural floor plan generation was proposed by Lopes et al. [14]. The method uses a grid representing the building allotment where the rooms are placed using an algorithm to determine restrictions such as adjacencies, connection and passage points and the functionality of each room. Taking this in account, the method expands the rooms in the grid. More related with the reconstruction of cultural heritage sites, Rodrigues and colleagues [15] focused their efforts on the reconstruction of roman houses. The starting step makes use of an L-System to generate the rooms. Then, based on Vitruvius conventions, adapted by Maciel [16], the authors wrote a set of rules responsible for guiding the automatic generation of the structures. The modeler was extended for modern houses.

An ontology-based tool was suggested by Liu et al. [17] for the reconstruction of the Chinese cultural heritage sites. In their system, the user can specify the rules of the building styles and an urban map to be reconstructed. The city generator module generates the city taking into account user's input and a defined ontology. Moreover, this module uses a checker to guarantee the coherency of the process, for example by ensuring that buildings are not created upon streets. To generate the buildings' virtual models the users use a grammar that contains the definition of these elements.

III. ERAS SYSTEM

ERAS is an expeditious virtual reconstruction tool, under development, for cultural heritage sites to serve archeological research purposes. The tool follows the structure suggested in [1], with two main processes, as it might be seen on Fig. 1: information extraction and procedural modeling. The information extraction process receives textual descriptions such as historical documents and interprets the containing information to pull out the relevant data, which will be used to fill-in a data model based on the defined ontology (also defined in ERAS scope). The referred filled data model contains definitions about buildings, building parts, divisions, transitions (doors and windows, for example), and roofs. This will feed, completely or partially, the procedural modeler which creates the virtual representation of these data. The modeler is also responsible for amplifying the unknown information about the building that is intended to be reconstructed. The procedural modeling process is done in three steps. First, the modeler generates the city plan, then it builds a simple model for each ancient city structure, and it finalizes the process with the generation of the buildings interiors and exteriors. The whole process relies on a set of rules, respecting the proposed ontology. The expected result is a faithful model provided for visualization and direct manipulation.

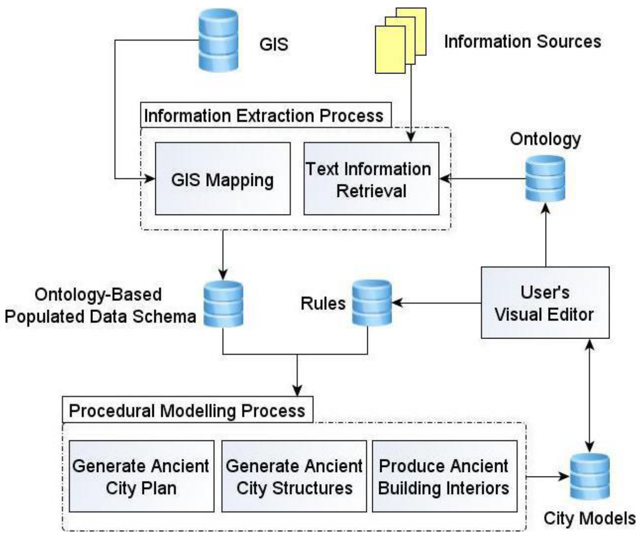

Figure 1. ERAS general architecure

IV. PROCEDURAL BUILDING GENERATION MODULE OVERVIEW

In this section the current solution for the last step of the procedural modeling process - the buildings generation - is presented. ERAS is intended to be an ontology-based tool for the expeditious reconstruction of ancient places, namely roman ancient structures. Currently, a methodology for reconstructing ancient roman houses – *domus* – is set up on the tool. The current procedural building generation module is composed by 3 main components: the ontology, the constraint rules and the procedural modeler. The ontology is related with the previous work [2] and defines a relationship between the generic elements that composes a city. It is established that a generic city is the top element, holding the blocks. These blocks can be seen as areas in where the other elements – buildings, gardens, city furniture - are aggregated. It is noteworthy that the procedural modeler's class diagram is based on this ontology, which establishes implicitly the rules for loading the incoming data. Presently, this data is stored in a text file just for testing the system. The text file contains definitions of how to divide the building area, how many building parts constitute the building, what rooms are present in each part, weight values for occupation, how they connect to each other and how to apply textures. The modeler uses a rule loader to gather information about the floor plan from the rules file. To generate the floor plan the modeler follows a squarified tree map strategy, which differs from the original algorithm in the way how the corridors are generated. To support the splitting tasks an L-System was defined. The L-System recursively queries the file in order to find replaceable divisions and the respective weights. The division process starts with the semantic term defining a building and its area. Then, the L-system searches in the file for the building term which works as an axiom to replace the older area by the new rectangular areas based on occupation weight. As a typical L-System, the process occurs until the last term is found in the file. At this point the modeler contains the definitions of the room positions and sizes. The next step is to connect the rooms to each other according to the functional zone specification defined in the rules. This results in a bidirectional graph that is used to place openings between rooms. The next step is to define the elements of connection with the exterior. One more time, the rules are queried to decide what rooms may contain window or what rooms hold a connection door to the exterior. The result is a floor plan that is extruded according to the determined doors and windows and the proper textures are set up. Finally, the roof is defined by uneven squares specified in the rules file.

V. ONTOLOGY INTEGRATION

The implementation of the methodology used to generate roman houses is regulated by the ontology previously defined [2] in which each element may correspond to a class. It was used a mechanism of inheritances and compositions in order to adapt the procedural modeler's class diagram, presented on Fig 2, to the ERAS ontology.

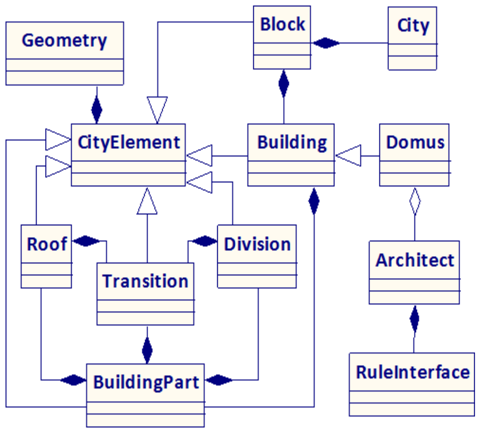

Figure 2. Procedural modeler's class diagram

The starting element, which contains all other, directly or indirectly, is the city. The city contains one or more blocks, each one holding the buildings. The buildings are composed by, at least, one building part, and each building part holds a set of divisions and a unique roof. Transitions are the connection points between divisions, building parts and roofs such as windows, doors and chimneys. In the case of a *domus*, a good example for roof transitions would be the *impluvium* that is a hole for collecting water. All the referred elements are a specialization from the City Element that contains geometry. Besides the referred ontology, the modeler has a rule interfacing class and an architect that employs a procedural technique which directly transforms the geometry in each element. In the next sections it will be explained the constraint rules, the L-System rules and the procedural technique used by the modeler.

VI. CONSTRAINT RULES

The methodology is guided by a set of rules that support the generation process of roman houses. They can be divided in two kinds of rules: The L-System rules and the constraint rules. The first ones are used to produce the rectangular areas for building parts and their respective rooms. They are used as a base to the first draft of the floor plan. The L-System will be explained in depth in the next section. The second set of rules stores information about room zone classification and connections between functional zones. The rooms are classified by functional zones in order to facilitate the association process between rooms (Fig. 3). Thus, instead of defining the allowed connections room by room, it is only needed to specify what groups are set up to connect, allowing a sort of flag system to create the transition points in the later generation process. This classification is intended to be transversal to every building in every architectonic style, maintaining, so far, the abstraction required for such extension.

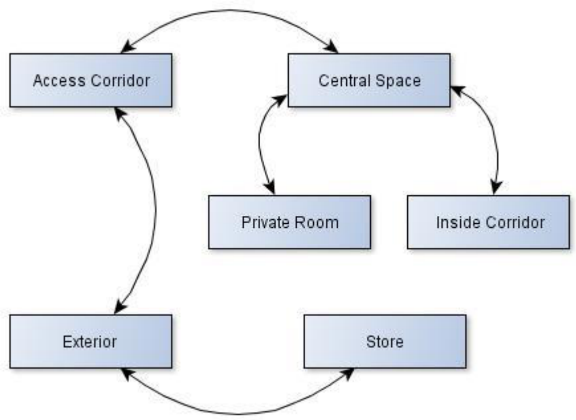

Figure 3. Representative schema of the connection between the functional zones

In the particular case of the roman houses, the rooms are classified as presented in table I. As it might be noticed, several functional zones only classify one room. Obviously, this happens because it is only being classified the rooms of one type of building. Ancient roman cities were composed by many other kinds of buildings such as public baths, theaters and temples. The desired resulting tool of ERAS extending to other buildings will increase hugely the number of building rooms, revealing the real utility of a classification system like this. There is also another kind of rules that can be perceived as auxiliary data. This data is used by the modeler to apply textures on the interior and exterior of the building, to generate the roof, to know relative measures such as room occupation area percentage or building part height and also to flag rooms connecting with the exterior, which determine if a window may be created or not. Next section will present the modeler and procedural technique that uses this rules to generate a building.

TABLE I. DOMUS ROOMS CLASSIFIED BY FUNCTIONAL ZONES

	Access Corridor	Central Space	Private Room	Inside Corridor	Store
Atrium		X			
Cubiculum			X		
Triclinium			X		
Tablinium				X	
Andron				X	
Peristylium		X			
Vestibulum	X				
Cullina			X		
Bathroom			X		
Oecus			X		
Exedra			X		
Alae			X		
Servae			X		
Tabernae					X

VII. SQUARIFIED TREEMAP ARCHITECT

In order to deal with the rules and apply them to transform the geometry of each element, it was developed a specialized class which can be seen as an architect. The class operates in two consecutive generation processes:

- The floor plan generation
- The extrusion based on the floor plan

A. The floor plan generation

The floor plan is generated in six steps. The first two steps are related to the Squarified Treemaps approach, generating the square areas for building parts and rooms (Fig. 4). This generation is guided by an L-System fed by the rules. The first iteration splits the initial area of the building into smaller squares which are the building parts. The areas of the building parts are then divided into smaller areas for the rooms. The splitting up, in each case, is made according to the relative weights (in percentage) defined by the rules. Instead of creating a connecting corridor in the end of the splitting process, as is proposed by the authors of the original Squarified Treemap technique, in the presented methodology the corridor is created during the splitting process, which guarantees that the rooms keep their areas and shapes. After the splitting up process is complete, it is created a graph to connect adjacent rooms according to the zone function classification present in the rules. This is used in the fourth step to mark the openings between the connected rooms. This step also inquires the rules to mark the windows that interface with the exterior. Next step alters the geometry of each room and each building part to include, properly, the marked doors and windows. The same step also applies hall thickness. The roof openings are also marked in this generation step. Some architectonic styles, as the roman one, connect architectonic elements to the roof, as is the case of the impluvium / compluvium. If such rule is present, the roof hole is pre-allocated in order to fulfill this architectonic convention. The result of these steps is a floor plan that is concordant with the rules defined to the architectonic style in use.

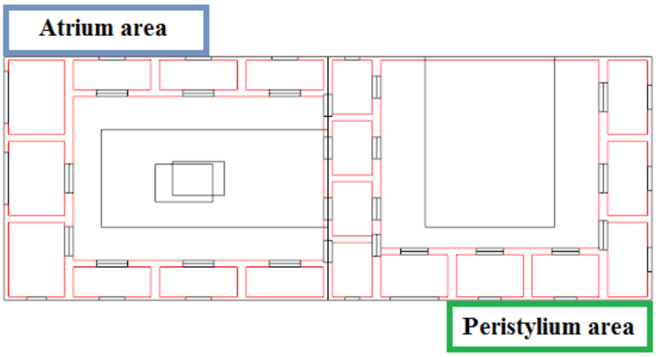

Figure 4. The floor plan generation, using a Squarified Treemaps approach: The two first squares generated by process are the atrium and peristylium area and the next iteration splits each areas in smaller squares for rooms.

B. *The Extrusion Process*

The procedural building generation module finishes its execution by extruding the resulting floor plan to produce a 3D model of the building. The process achieves its objective in a few steps. First of all, the walls height of the rooms and building parts are increased in the places where do not exist openings. Then, the wall portions around the openings are properly extruded. In both cases, the texture is applied. Again, the rules are guiding the process by providing informations like building part height, door maximum height, windows framing and textures of each element. At last, the roof is defined. The modeler consults the rules that define a set of multilayer rectangles. Each rectangle is at a specified height and, relatively to the ground parallel plan, each present rectangle is contained in the previous one. The vertexes of each rectangle are connected to the next one following the order of the indexes and a texture is applied, finishing the process of extrusion. Some results are shown on Fig. 5.

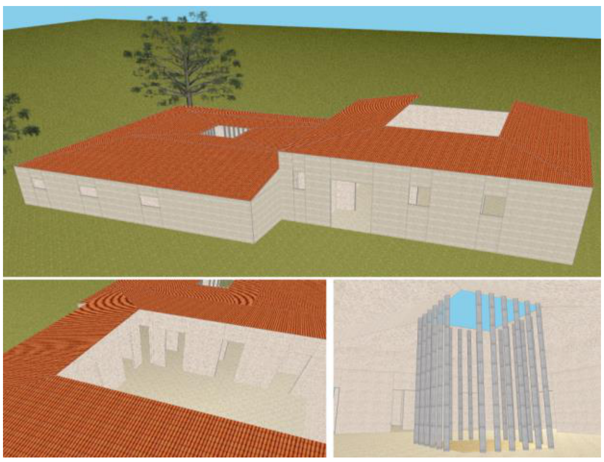

Figure 5. Final layout of a generated domus from the outside and from the inside

VIII. CONCLUSIONS AND FUTURE WORK

Presently, ERAS ontology-based tool for procedural generation of ancient structures implements a working methodology to generate roman houses that integrates a full ontological specification of an abstract city with the possibility to support other architectonic styles. The methodology also relies in a set of rules to generate properly the structures that so far has pointed out that the ontology, acting as a regulation part of the system, is being developed in the right direction. On the other hand, the procedural modeler connects successfully with the rules and transforms them in the proper 3D model. In the serious games context, the models produced by the modeler may be an asset to teach history or to research in the archeology area, by including the architectonic knowledge left by Vitruvius documentation [16] and consequently ensuring, at least, some degree of realism. However, this pilot prototype capable of producing a few derivations of roman houses needs to be later validated by archeology experts in order to improve the proposed methodology and related techniques. To test the system we plan to conduct a series of validation studies with archaeologists. Several models will be generated by our system and these models will be presented to the archaeologists. After a period of analysis they will answer to a questionnaire where they validate the models against a set of parameters, that will include aspects such as realism and historical accuracy and fidelity. However, this work needs to be improved in order to develop a more solid set of rules contemplating new situations in the generation, such as windows side by side in neighbor buildings or the structural organization of other types of buildings. The procedural modeler may also integrate other procedural techniques besides the Squarified Treemaps in order to adapt the generation to non exclusive rectangular footprints. By including these features in the present modeler, it is intended to achieve a full procedural system capable of generating coherent structures described by multi shaped geometry, constituting a whole archaeological site with an improved visual accuracy.

ACKNOWLEDGMENT

This work is partially supported by the Portuguese government, through the National Foundation for Science and Technology – FCT (Fundação para a Ciência e Tecnologia) and the European Union (COMPETE, FEDER) through the project PTDC/EIA-EIA/ 114868/2009 entitled “ERAS - Expeditious Reconstruction of Virtual Cultural Heritage Sites”.

REFERENCES

[1] T. Adão, L. Magalhães, M. Bessa, J. Barreira, M. Melo, M. Gonçalves, J. Sousa, and E. Peres, “Proposal of an Information System for a Semi-automatic Virtual Reconstruction of Archeological Sites,” presented at the 4th Conference of ENTERprise Information Systems, Algarve, Portugal, 2012.

[2] T. Adão, L. Magalhães, M. Bessa, A. Coelho, A. Sousa, N. Rodrigues, R. Rodrigues, F. Pereira, J. Moura, and L. Reis, “ERAS – An Ontology-Based Tool for Expeditious Reconstruction of Virtual Cultural Heritage Sites,” presented at the Actas do 20º Encontro Português de Computação Gráfica, Viana do Castelo, Portugal, 2012.

[3] A. Lindenmayer, “Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs,” Journal of Theoretical Biology, vol. 18, pp. 300-315, 1968.

[4] Y. I. H. Parish and P. Müller, “Procedural modeling of cities,” presented at the Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001.

[5] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, “Instant architecture,” presented at the ACM SIGGRAPH 2003 Papers, San Diego, California, 2003.

[6] P. Müller, T. Vereenooghe, P. Wonka, I. Paap, and L. Gool, “Procedural 3D Reconstruction of Puuc Buildings in Xkipche,” in Proceedings of Eurographics Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 06), 2006, pp. 139–146.

[7] K. Dylla, P. Müller, A. Ulmer, S. Haegler, and B. Frischer, “Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques,” presented at the Making History Interactive. 37th Proceedings of the CAA Conference, Williamsburg, Virginia, 2009.

[8] A. Rau-Chaplin, B. Mackay-Lyons, and P. Spierenburg, “The Lahave house project: Towards an automated architectural design service,” in Proc. Int. Conf. Comput.-Aided Design, Hagenberg, Austria, 1996, pp. 25–31.

[9] J. Martin, “Procedural House Generation: A method for dynamically generating floor plans,” in Symposium on Interactive 3D Graphics and Games, ed, 2006.

[10] E. Hahn, P. Bose, and A. Whitehead, “Lazy Generation of Building Interiors in Realtime,” in Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on, 2006, pp. 2441-2444.

[11] T. Tutenel, R. Bidarra, and R. Smelik, “Rule-based layout solving and its application to procedural interior generation,” in CASA Workshop 3D Adv. Media Gaming Simul, Amsterdam, The Netherlands,, 2009, pp. 15–24.

[12] F. Marson and S. R. Musse, “Automatic Real-Time Generation of Floor Plans Based on Squarified Treemaps Algorithm,” International Journal of Computer Games Technology, vol. 2010, 2010.

[13] E. Schkufza, P. Merrell and V. Koltun, “Computer-generated residential building layouts," ACM Trans. Graph., vol. 29, pp. 1-12, 2010.

[14] R. Lopes, T. Tutenel, R. M. Smelik, K. J. d. Kraker, and R. Bidarra, “A constrained growth method for procedural floor plan generation,” presented at the Proc. 11th Int. Conf. Intell. Games Simul., 2010.

[15] N. Rodrigues, M. Dionísio, A. Gonçalves, L. Magalhães, J. P. Moura, and A. Chalmers, “Rule-based Generation of Houses,” Computer Graphics & Geometry, vol. 10, 2008.

[16] M. Maciel, Vitrúvio - Tratado De Arquitetura: Ist Press, 2006.

[17] Y. Liu, Xu, C., Zhang, Q. e Pan, Y., “The Smart Architect: Scalable Ontology-Based Modeling of Ancient Chinese Architectures,” Intelligent Systems, IEEE, vol. 23, pp. 49-56, 2008.