Wood Mass-Customized Housing

A dual computer implementation design strategy

Krystian Kwiecinski[1], Filipe Santos[2], Ana de Almeida[3], Bruno Taborda[4], Sara Eloy[5]
[1] Warsaw University of Technology (WAPW) [2,5] Instituto Universitário de Lisboa (ISCTE-IUL, ISTAR-IUL) [3] Instituto Universitário de Lisboa (ISCTE-IUL, CISUC)
[4] Instituto Universitário de Lisboa
[1] krystian.kwiecinski@arch.pw.edu.pl
[2,3,4,5] {filipe.santos|ana.almeida|bruno_taborda|sara.eloy}@iscte.pt

This paper reports our current research on automatic generation of houses layouts according to future inhabitant's requirements. For that generation we propose the use of a design method based on shape grammars that encodes light wood frame construction guidelines. Two different implementations for the design system are presently under development. One based on shape grammars supplemented with procedural knowledge and another using a genetic algorithm. Both implementations allow the generation of house layouts that fulfill both the user requirements and the design language.

Keywords: *shape grammar, genetic algorithm, computer implementation*

INTRODUCTION

This paper will focus on an undergoing research that aims at developing a system that allows housing future inhabitants to participate in the design of their house by using mass-customization design (Kwieciński and Slyk 2014). The proposed system is based on shape grammars and will allow future owners to acquire houses that fit their needs while simultaneously complying to a language of design (Stiny 1980). Such a system will enable to deliver design solutions with quality (Eloy et al. s.d.) and at affordable prices improving satisfaction of clients. The system encodes knowledge on housing design principles (types and sizes of rooms and preferable connections) as well as in the timber construction technique that underlies the shape composition principles.

To make the system available for future inhabitants a computer implementation is being developed so that it could be made available online or at the housing construction company. The design tool will: i) deliver design solutions answering the client's' requirements based on a design grammar therefore maintaining a language of design, ii) be feasible to be used by non designers (e.g. clients, sales staff of a construction company). With the aim of finding a good technical solution to satisfy users' needs two strategies of implementation are being developed simultaneously and in close collaboration. One is based in shape grammars supplemented with processes and the other based in genetic algorithms.

Research in mass-customized design using shape grammars have been developed over the years and several shape grammars have addressed the housing problem and the need to make available design systems that respond to the inhabitants needs (Duarte 2001; Colakoglu 2005; Benros et al. 2011; Eloy 2012; Coimbra and Romão 2013). The main difficulty to fully use these systems in design practice is the delay of the computer implementation of architectural design processes, namely by the use of shape grammar logics. Computerized design tools that offer design alternatives supporting the design ways of thinking and working are still rare.

Current systems allow some aspects of generation of shape and are based on a built-in drawing editor, e.g. Rabo-de-Bacalhau grammar tool (Strobbe 2015) or in links to a CAD software, e.g. Grappa (Grasl 2012) for Revit (McKay et al. 2012). These systems are essentially academic based and not yet aiming at being used in the real design scenario. Design applications like Flemming (1987) Queen Anne Houses tool, Li (2002) Yingzao fashi grammar, Grasl (2012) Grappa tool and Ligler and Economou (2015) Entelechy Grammar are systems that allow to develop designs according to languages of past styles. Those tools are used specially in an historical and design understanding perspective and not as tools to design new solutions for architecture problems. Other implementations like Strobbe (2014) Rabo-de-Bacalhau grammar tool allow to generate new design solutions that can then be used by the house owners to do their refurbishment. The goal of the presented research is to deliver such a tool that can deliver designs able to be immediately used as a design project to build a house.

Catalogues of house designs are very popular in several countries. Future owners are provided a catalogue of standardized solutions for them to choose but those solutions are restricted to a small number of changes and the act of choosing is based on looking to all solutions and not to the ones that may respond to the clients need. In this research we aim at looking for systems that allow users to play with a house design tool allowing them to find the best fit house to their own personal wishes.

WOOD MASS-CUSTOMIZED HOUSING

Choice of the house construction system has been determined by its popularity in some european countries as well as the possibility of its prefabrication off site and partial automation of manufacturing process. In order to facilitate mass-customization chosen construction system have to enable construction flexibility and permit just-in-time production. The use of industrialized methods of production allows to save time and permit cheaper costs and quality control. For these reasons a light wood frame construction was chosen as the main structure of the building. In consequence developed shape grammar reflects the capabilities and constraints of the chosen building system.

Chosen building technique uses vertical and horizontal structural members, studs, which provide a stable frame to which interior and exterior wall coverings are attached. The structures are strengthened with rigid panels of oriented strand board (OSB) used to form wall sections. Sections of the walls can be prefabricated off-site therefore the whole building could be erected in just a few weeks.

Wall framing in light frame construction system includes the vertical and horizontal members of exterior walls and interior partitions (see Figure 1). The walls are build with the use of standardized studs which are placed in the axial spacing of 60 cm. This structural guideline is encoded in the shape grammar rules. Therefore the shape rules are operating on rooms which dimensions are a multiplication of 60x60 cm module. Moreover the maximum permissible span of the ceiling, up to 6 m, affects the choice of the house typology, therefore also the localization of load bearing walls as well as the width of the building in general and rooms in particular.

The possible design solutions include detached houses that can have either one or two floors, a compact rectangular floor plan, where only the garage sticks out of the rectangular perimeter. The houses have a fixed width, a center axis with a right and left side of the same width, and the house length may vary. The position of the houses may be parallel or perpendicular to the access route. For the purpose of the computer implementations presented in this paper the scope of possible design solutions was narrowed down to single floor houses located perpendicular to the access route.

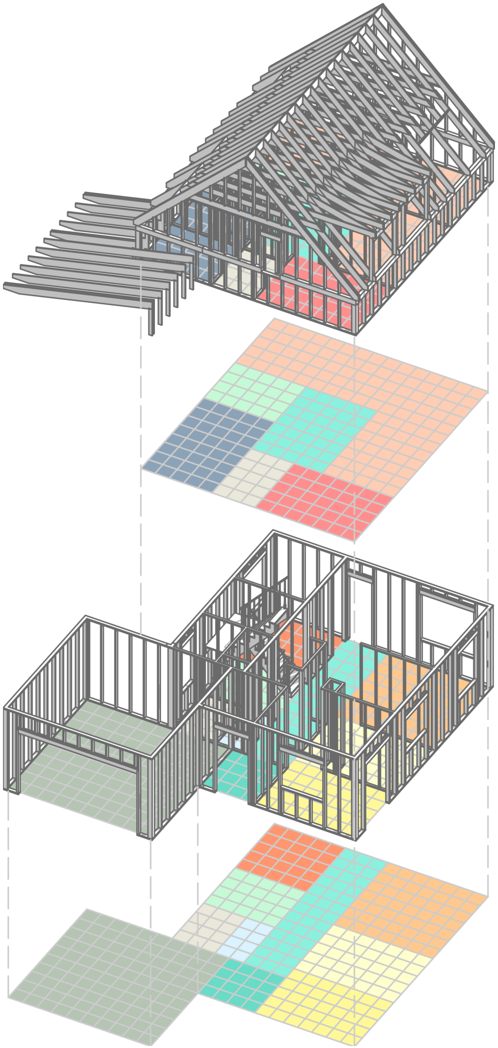

Figure 1
Rooms modularity resulting from light wood frame structural guidlines.

SHAPE-GRAMMAR

The proposed grammar consists of a vocabulary of shapes and shape rules and a generation process responsible for the generation of the layout solutions. Shape rules have a shape part and a condition part which enable the application of rules preventing the application if some conditions are not fulfill. Rules are grouped in order to differentiate 4 design stages (see Figure 2): generation of the the grid, generation of rooms belonging to the entrance zone, generation of rooms belonging to the semi-public zone and generation of rooms belonging to the private zone. Each stage ends after all the required rooms of that stage are generated leading the generation to the next stage. The whole process is finalised with the generation of a layout where all the required rooms are present and all the conditions are fulfilled. Future owners may then see the possible design solutions for their requirements and decide which they want to opt for.

The generation process initiates by collecting information from the future inhabitant in order to formulate the house design brief. At this stage of the development the users are asked to specify the number of future inhabitants and the quality level of the house (minimum, recommended or optimum). Based on that information the size of the house is calculated and the functional program of the building is specified which can be further modified by the user. Besides the layout design that is focused in this paper, the future inhabitant will be able to modify all other aspects of architecture as where to build, what construction system and materials to use, among others.

In the first stage of the house derivation a grid of 60x60cm cells is generated, which reflects the chosen size of the house. In the next stages the rooms belonging to different building zones are placed. The placement of rooms occurs in relation to external factors (e.g. next to the front wall or next to the back wall) and internal context (e.g a room is placed next to another one). Conditions for grouping the rooms constrain the layout configurations and guarantees

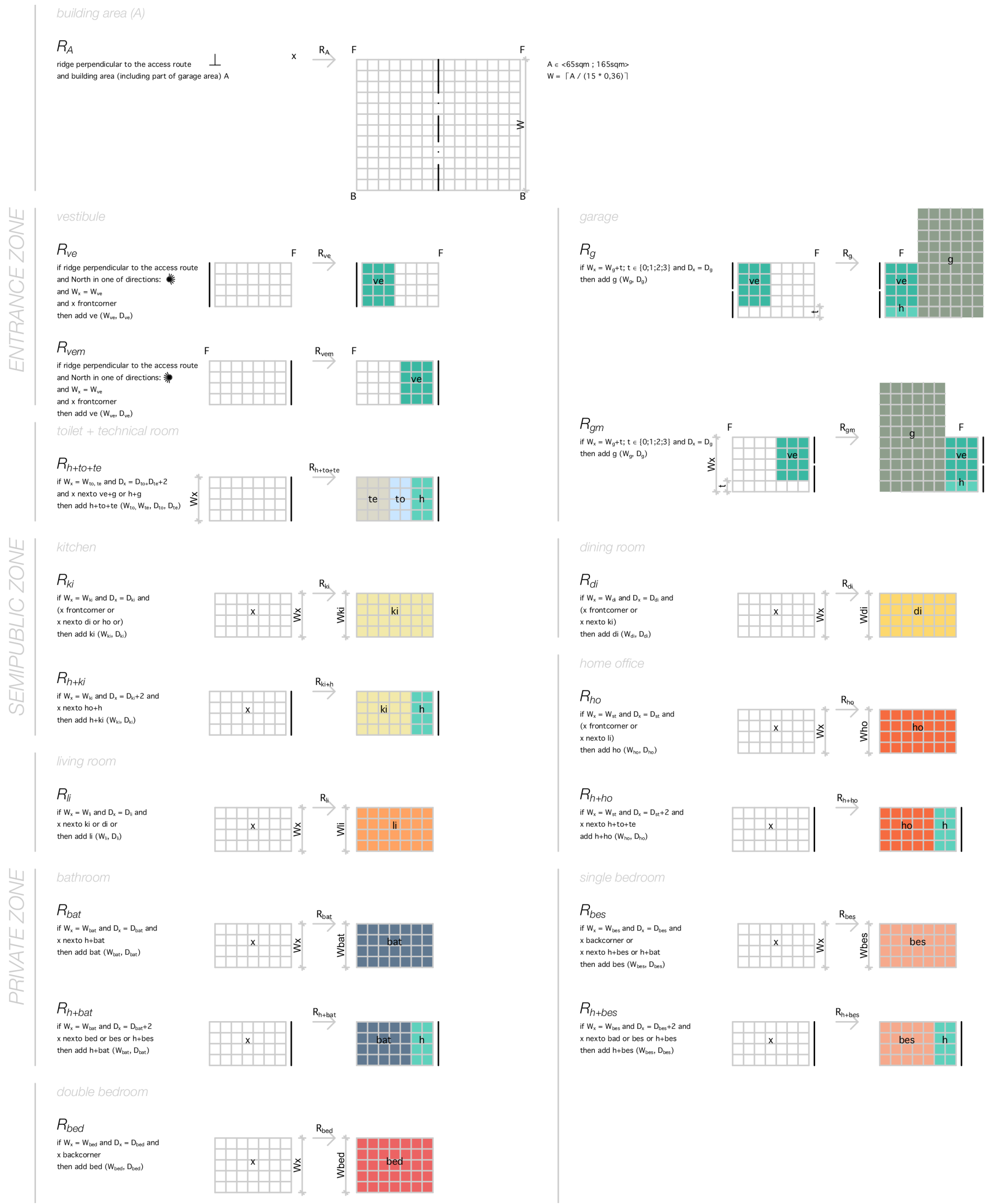

Figure 2
Shape rules.

the correct placement of rooms according to housing design standards. In order to preserve characteristics as privacy, conditions for placing rooms are defined which suggests e.g. placing the social areas closer to the entrance and the private ones deeper in the house.

The developed grammar uses the algebra U12 with bi-dimensional, linear base, floor plan of the building. Automatically generated house layout solutions based on client's inputs and a set of design rules enable clients to participate in the design process by acquiring only the designs that meet their goals and requirements. Proposed grammar by providing several design solutions, allows future inhabitants to search for the ones that fits their needs. Generated solutions can be compared with each other or might lead to the decision on changing the design brief and generating new solutions fulfilling modified user requirements.

IMPLEMENTATION

Two different implementations for the design system are presently under development. One based on shape grammars supplemented with processes detaches shape knowledge from procedural knowledge. The exponential nature of the search space led to the need of employing different effective search paradigms such as genetic algorithms which are used in a second implementation.

Shape Grammar Implementation

The shape grammar implementation is based on (Santos and Esmerado 2015) where differences with respect to conventional shape grammars consist on the adoption of any primitive vocabulary of shapes and a complete detachment between shapes and processes. The shape part is conceived to meet the practical objectives of the design system and uses a simplified form of sub-shape detection. Shapes are represented by identifiers associated with pictures and positioned in a coordinate system referring additionally to the applied Euclidian transformation. Shape compositions are represented by sets of shapes positioned in the same coordinate system. Processes use the usual procedural notions for capturing sequences, alternatives and property tests for the application of shape rules and conditional shape rules during the design process.

To give an idea of the computational model used in this shape grammar implementation, for producing houses layout design solutions, the following basic shapes (see Figure 3) are considered to represent front and back markers and 0.6x0.6 m cells with different room occupation:

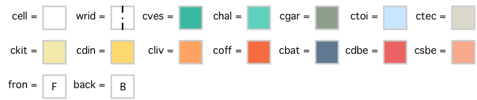

Figure 3
Basic shapes.

Each identifier represents the picture positioned in the respective coordinate system xOy.

Positioning a shape defined in this way within a different coordinate system x'O'y' is represented by a pair (shape, transformation) specifying the transformation (translation, rotation, scale, etc.) required for positioning the coordinate system xOy associated to the defined shape into the new coordinate system x'O'y'.

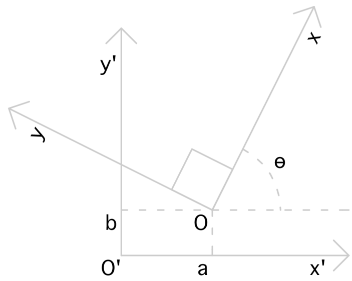

Figure 4
Positioning cell within a new coordinate system.

For instance, in Figure 4, cell is positioned into the coordinate system x'O'y' by a scale S(c,d) and rotation R(q) followed by a translation T(a,b) and thus is represented by (cell, T(a,b)R(q)S(c,d))).

Using these basic shape representations, the following shape compositions (Figure 5)

Figure 5
An exemplary shape composition.

are represented respectively by:

```
basement3x3 = {(cell,Id), (cell,T
 ↪ (0.6,0)), (cell,T(1.2,0)), (cell,T
 ↪ (0,0.6)), (cell,T(0.6,0.6)), (cell
 ↪ ,T(1.2,0.6)), (cell,T(0,1.2)), (
 ↪ cell,T(0.6,1.2)), (cell,T(1.2,1.2)
 ↪ )}, vestibule3x3 = {(cves,Id), (
 ↪ cves,T(0.6,0)), (cves,T(1.2,0)), (
 ↪ cves,T(0,0.6)), (cves,T(0.6,0.6)),
 ↪  (cves,T(1.2,0.6)), (cves,T(0,1.2)
 ↪ ), (cves,T(0.6,1.2)), (cves,T
 ↪ (1.2,1.2))}.
```

Conditional Shape Grammar Rules are pairs of shape compositions plus a condition allowing it application. The following shape grammar rule (see Figure 6) is represented by placeVestible with precondition (next(matched(basement3x3),fron), basement3x3, vestibule3x3).

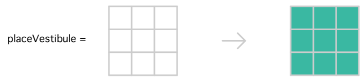

Figure 6
An exemplary conditional shape rule.

Using identifiers for shape grammar rules allow its reference in processes describing the design development. Processes may also have names allowing it reuse in other processes. The following processes describe the steps for producing houses layout design solutions:

```
home = placeBasement; placeEntrance;
 ↪ placeSemipublic; placePrivate ;
 ↪ verify(homeRestrictions).
placeEntrance = placeVestibule;
 ↪ placeGarage;
 ↪ placeTechRoomandToilet.
placeSemipublic = placeHomeOffice and
 ↪ placeKitchen and placeDiningRoom
 ↪ and placeLiving Room.
placePrivate = placeDoubleBedroom and
 ↪ placeSingleBedrooms and
 ↪ placeBathroom.
placeSingleBedRooms =
 ↪ placeSingleBedRoom1 or
 ↪ placeSingleBedRoom2 or (
 ↪ placeSingleBedRoom1 and
 ↪ placeSingleBedRoom2).
homeRestrictions = (next(vestibule,
 ↪ garage) && next(toilet,garage) &&
 ↪ next(technicalRoom,garage) && (
 ↪ next(fron,homeOfice) || next(fron,
 ↪ homeOfice) || next(fron,homeOfice)
 ↪ ) && (next(diningRoom,fron) ||
 ↪ next(dinningRoom,Kitchen)) && (
 ↪ next(homeOfice,fron) || next(
 ↪ homeOfice,livingRoom)) && (next(
 ↪ bathRoom,doubleBedroom) || next(
 ↪ bathRoom,singleBedroom)) && some(
 ↪ doubleBedroom,next(doubleBedroom,
 ↪ back)) && some(singleBedroom,next(
 ↪ singleBedroom,back)) && (next(
 ↪ kitchen,fron) || next(kitchen,
 ↪ dinningRoom) || next(kitchen,
 ↪ homeOfice)) && (next(livingRoom,
 ↪ fron) || next(livingRoom,kitchen)
 ↪ || next(livingRoom,dinningRoom)).
```

The shape grammar rules and the alternative compositions of the process offer the possibility to generate a shape composition among different alternatives and thus a non-deterministic perspective is followed in the design process. Operationally, the design process applies repeatedly shape rules to the intermediate shapes obtained at each step according to the sequence order established in the process. Due to non-determinism, each application of the process to an initial shape composition may produce different designs by forward chaining using some operational preference in the choice of the alternatives. Each time a test process fails or a shape grammar rule fails to apply, the system backwards trying to build a different solution.

The tree of Figure 7 exemplifies the generation process of houses layout design included in circles fulfilling the design rules present so far for a 17x14 module house with the following rooms and dimensions: 3x3 vestibule, 3x3 technical room, 9x6 garage, 3x2 toilet, 3x3 storage, 2x7 kitchen, 4x7 dining room, 5x7 living room, 4x7 double bedroom, 4x7 single bedroom, 4x7 single bedroom with hall and 3x5

Figure 7
An example of the generation process of house layout design.

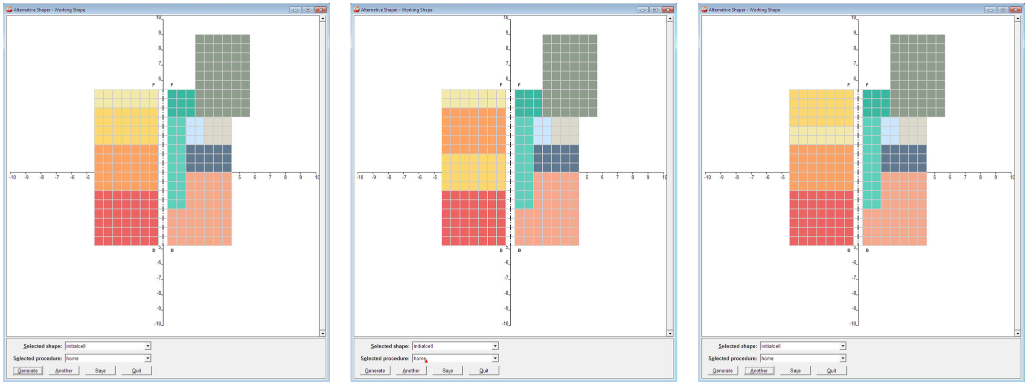

Figure 8
Alternative Shaper Workbench.

bathroom.

A workbench "Alternative Shaper" supporting this approach has been developed for exploring diverse case studies and is actually adapted to implement the referred design system (see Figure 8).

This first tool was implemented in SWI-prolog in order to explore a depth first search strategy of the design solution space.

Not only non-determinism of the procedural part is intensively used to explore the design solution space but also is used to describe sketchy and incomplete design processes on the specification level. As a result, this approach allows also design specifications emphasizing design properties instead of processes.

Genetic Algorithms Implementation

Evolutionary algorithms have been positioning as flexible and useful tools within diversified research areas. Based on the Theory of Evolution, more exactly on the notion of the survival of the fittest, Genetic Algorithms (GA) are able to search for floorplan design solutions, evaluate them and continue the search for the better fitted floor plans based upon that evaluation. We propose an approach where the evaluation criteria are the representation of the generative shape rules considering the general positioning of the house required rooms. A prototype tool, ShaperGA, using a genetic algorithm to generate several floor plan design alternatives is presented.

In GA terms, a solution to a problem is an individual or chromosome, the existent solutions at each stage are grouped as a population, and the creation of a new population is a generation. Each individual, representing a house layout, is a group (array) of binary matrices, the genes, each of these representing a house division and its actual positioning in the grid. The fitness of an individual corresponds to the value of the objective function that is evaluated in terms of the floor plan positioning rule's compliance. The three classical genetic operators - selective reproduction, crossover and mutation - are used for design evolution. The GA starts with a random generation of an initial population, evaluates the individual designs created, applies genetic operators to (some of) these individuals, producing a new generation of floor plans, and loops until termination. Such a tool enables effective automatic exploration of the design space, evolving floor plans already found according to the predefined criteria towards the desired goal.

Evolutionary computation as a technique is not new to architectural design, specially for creative generative design tools (Caldas and Norford 2002; Steadman 2008; Knecht and König 2010; Pasternak and Kwiecinski 2015). The GA is here employed as a generative tool with evolution guided by a fitness function that embodies the shape grammar rules,

Figure 9
Genetic Algorithm Implementation.

thus measuring the aptitude of a generated layout in terms of rules' fulfillments. The areas and rooms relative positioning rules are incorporated into the fitness function and the violation of any of them results in a fitness penalization. The algorithm also needs the specific user information on the floor plan to be generated, namely, which rooms and side dimensions are to be inserted to begin the generative process and design evolution towards feasible layouts.

This approach, Shaper-GA, begins by randomly generating individuals (layouts) where each comprises the predefined needed rooms. However, being random, most of these layouts will not be feasible because of overlaps or rules' failure of compliance. Through the referred incorporation of penalties in the fitness function and usage of this knowledge in parental selection and for reproduction, the best fitted individuals (houses' layouts that best follow the design rules) are selected and their genes propagated to the offspring. From the overall set of parents and their offspring, the better fitted houses are chosen for the next generation, Thus, the repeated process evolves the better layouts until being able to generate completely feasible floor plans.

Shaper-GA has undertaken the necessary trial experiments with fixed specifications much alike the example described in the previous section: all the required rooms have fixed dimensions, and thus also fixed orientations. The experimentation results are very favorable, with Shaper-GA evolving to feasible floor plans (two examples on the left-hand side of Figure 9) within reasonable computational time. The visual interface will display all of the feasible layouts (left and right arrows at the bottom of Figure 9). Also any of the remaining (lesser) fitted solutions found in the final evolutionary generation can be visualized (example on the right-hand side of Figure 9). The interface is prepared to show product layouts where some of the relative rooms' positioning rules have been relaxed thus allowing for the emergence of other designs where the user might see fit.

RESULT AND DISCUSSION

Both mentioned implementations are actually supporting test experiments for generating house layouts that comply with a language of design and respond to user requirements. Up to now, for small houses' layouts, the process shape grammar implementation produces solutions in just a few seconds, faster than the genetic algorithm shape grammar implementation. However, the number of potential solutions is combinatorial on the number of divisions and their side dimensions. Thus, when the number of divisions increase and their sizes change, we are expecting that the process shape grammar implementation will take a considerable more amount of time just to produce a solution. That's where the genetic algorithm counterpart plays an important role.

Shaper-GA implementation is being upgraded to allow for evolutionary readjustment of the rooms dimensions, according to predefined interval range values. Meanwhile, the best way to integrate the connections between rooms into the evolutionary generation is also being studied. More research will be done in order to develop fully efficient prototype software for both approaches. For that we'll be using heuristics to find paths of promising generating designs and speeding up the algorithm so that several optima are quickly found. One additional approach will also be explored for the cases where there are non-optimal solutions, i.e. there are no solutions satisfying all the shape conditions. For that situation the genetic algorithm approach will be used to integrate a multi-criteria representation of customer's preferences.

The results found so far, make us believe that these generative tools are promising ones that satisfy the users' needs and can be used by non designers in order to explore housing design solutions.

ACKNOWLEDGEMENTS

This work was partially supported by Center For Advanced Studies Warsaw University of Technology through Outgoing Scholarship No. CAS/37/POKL.

REFERENCES

Benros, D, Granadero, V, Duarte, JP and Knight, T 2011, 'Automated Design and Delivery of Relief Housing: The Case of post-Earthquake Haiti', *CAAD Futures: Designing Together*, pp. 247-263

Caldas, LG and Norford, LK 2002, 'A design optimization tool based on a genetic algorithm', *Automation in Construction*, 11(2), pp. 173-184

Coimbra, E and Romao, L 2013 'The Rehabilitation Design Process of the Bourgeois House of Oporto: Shape Grammar Simplification', *eCAADe 2013: Computation and Performance*

Colakoglu, B 2005, 'Design by grammar: an interpretation and generation of vernacular hayat houses in contemporary context', *Environment and Planning B: Planning and Design*, 32(1), pp. 141-149

Duarte, JP 2001, *Customizing mass housing: a discursive grammar for Siza*, Ph.D. Thesis, Massachusetts Institute of Technology

Eloy, S 2012, *A transformation grammar-based methodology for housing rehabilitation*, Ph.D. Thesis, Universidade Técnica de Lisboa

Eloy, S, Vermaas, P and Andrade, M (s.d.), 'The Quality of Designs by Shape Grammar Systems and Architects: A Comparative Test on Refurbishing Lisbon', *Journal of Architecture and Planning Research*, (forthcoming)

Flemming, U 1987, 'More than the sum of parts: the grammar of Queen Anne houses', *Environment and Planning B: Planning and Design*, 14(3), pp. 323-350

Grasl, T 2012, 'Transformational Palladians', *Environment and Planning B: Planning and Design*, 39(1), pp. 83-95

Knecht, K and Konig, R 2010, 'Generating Floor Plan Layouts with K-d Trees and Evolutionary Algorithms', *Generative Art Conference*, , pp. 239-253

Kwiecinski, K and Slyk, J 2014 'System for customer participation in the design process of mass-customized houses', *Fusion: Data Integration at its Best, Vol 2*, pp. 207-215

I-kang Li, A 2002 'A prototype interactive simulated shape grammar', *Design education: Connecting the Real and the Virtual*, pp. 314-317

Ligler, H and Economou, A 2015 'Entelechy I: Towards a Formal Specification of John Portman's Domestic Architecture', *Real Time - Extending the Reach of Computation, Volume 1*, pp. 445-452

McKay, A, Chase, S, Shea, K and Chau, HH 2012, 'Spatial grammar implementation: From theory to useable software', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(02), pp. 143-159

Pasternak, A and Kwiecinski, K 2015 'High-rise Building Optimization A Design Studio Curriculum', *Real Time - Extending the Reach of Computation, Volume 1*, pp. 305-314

Santos, F and Esmerado, J 2015, 'A different shape grammar approach for automatic design generation', *International Journal of Advances in Computer Science & Its Applications*, 1, pp. 90-97

Steadman, P 2008, *The Evolution of Designs: Biological Analogy in Architecture and the Applied Arts*, Routledge

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

Strobbe, T 2015, *Computer-Aided Exploration of Architectural Design Spaces: a Digital Sketchbook*, Ph.D. Thesis, University of Gent