The Game of Urban Attractiveness

Shape Grammars and Cellular Automata Based Tool for Prediction of Human’s Behaviour in Cities

Paweł Unger¹, Luís Romão²
¹Gdańsk University of Technology, Poland ²University of Lisbon, Portugal
¹pawel.unger1@gmail.com ²lromao@fa.ulisboa.pt

This paper presents a way to predict people's interest in a public space based on a space's "attractiveness" as a movement attractor. Two generative systems are integrated into the prediction model. The Cellular Automata (CA) is the core of simulation engine and the Shape Grammars (SG) is a descriptive language for the CA rules. Both, CA and SG exhibit complementary features counteracting each other's drawbacks. Having translated social behaviour into a set of rules, the CA algorithm applies them to distinguish people's leisure interest attractors from places with a minor attractiveness. The tool is designed to be used at various urban scales by city planners and venture capitalists. It is dedicated towards the early stage of planning process to evaluate the future attractiveness of places. The case study is located in the central district of Lisbon, Bairro Alto. One of the important aspects are description of the rules with SG and interpretation of the CA results. Implemented in Python for Grasshopper and visualised in Rhinoceros3D. The article does not present the final solution, rather is an experimental attempt to interpret and describe the already explored urban context of Cellular Automata.

Keywords: Behaviour Prediction, Cellular Automata, Shape Grammars, Space Attractiveness, Urban Simulation

OVERVIEW
Cities are multi-factor structures that are possible to analyse, yet challenging to predict. Having information about presumed attractors of people's interest in public space would give various possibilities for interpretation. Knowing future attractive spaces in a city would examine the efficiency of public transport and the local zoning plan or would benefit the targeting of new investments. The tool may be used in various scales from the regional and urban to architectural. Technically, with major adjustments, it could be used on country or global scale. The case study is focused on a part of a Bairro Alto, a district of Lisbon.

CELLULAR AUTOMATA AND SHAPE GRAMMARS CONTEXT
Prediction of human behaviour has a long history and is one of the unfinished research topics to be
explored during decades. Cellular Automata (CA) model has been investigated in the crowd behaviour context since 90s (Dudek-Dyduch and Wąs 2006). Initially, the researchers focused on car traffic, then pedestrian dynamics prediction in practice (Fukui and Ishibashi 1999; Burstedde et al. 2001; Dijkstra 2000). Former experiments show that human mobility can be predicted with a 93% accuracy rate (Song 2010). Yet that method was not CA based and relied on tedious collection of data location from 50 000 cell phones to monitor the behaviour. To contextualize CA, in 1998 the environmental Research Institution of the Dutch government used the first CA-based integrated model to assist planners and policy makers for new urban planning strategies (White and Engelen 2000). The LeefOmgevingsVerkenner (Environment Explorer) is an integrated model of land use and the regional distribution of population and economic activity in The Netherlands (White and Engelen 2000). This model was used on a country scale.

Shape Grammars is a powerful tool for shape generation. The model consists of sets of shape, shape relations, shape symbols or labels and, most importantly, the shape rule to execute while a specific conditions allow its application (Stiny 1980). While researching behaviour simulation with SG, plenty examples on simulating the logic of compositions are noted (Durate 2005; Stiny and Mitchell 1978) or sole human behaviour (Motalebi et al. 2017). Also, urban development based on the SG models were designed (Durante et al. 2007). This paper will advance the idea of connecting Shape Grammars and Cellular Automaton to predict the human attractiveness distribution in public space.

**USING CA WITH SG**
The method is based on G. Stiny’s Shape Grammars (SG) concept and Cellular Automata (CA) model from S. Ulam and J. V. Neumann. Both systems have pros and cons but reveal a surprising compatibility and complement each other. Both concept strives to recognize patterns and apply transformations regarding conditions. SG uses transformation rules and start the transformation process from the initial shape. Rules are applied manually or by machine, always as sequentially determined while CA runs out a set of rules automatically and recursively from the initial shape (Speller et al. 2007). After the iterative process, SG design allows the analysis, explanation, and evaluation. In the SG sequence the approach is slow but determined by the designer. The CA recursive nature is fast but does not allow changes during analysis process. CA is easy to use for design generation but too abstract and productively unrestrained. Choosing self-generating rules is beyond human capability (Speller et al. 2007). SG is simple to perceive and conceive but manually laborious (Speller et al. 2007).

Similarities. Both provide an ease of design and visualization. Essentially, one system can complement the weakness of the other to stand for fast behavioural genomic system (Speller et al. 2007). In other words, SG is an easily understandable model to explain the meaning of the rules, which helps describing a particular rule. Hence CA is selected as the “engine” to accelerate the iterative process and to bring about acceptable performance.

**DEFINING PARTS OF CA WITH SG**
The authors added a 6th element to the list of the five essential definitions to describe CA (White and Engelen 2000):

- “A topography of cells,
- a state which characterizes the grid cells
- a definition of the closest neighbourhood of a cell,
- a set of transition rules that determine the state transformation of each cell,
- a sequence of discrete time step with all cells updated simultaneously” (White and Engelen 2000).
- an initial state of cells in the working area.

Now, the CA parts will be explained in the SG vocabulary.
The grid space is typically assumed to be two-dimensional, rectilinear and homogeneous (White and Engelen 2000). Alternatively, non-regular grids can be used, as the cadastral one used by Stevens and Dragičević (2007). 3D public spaces are exceptions in cities, yet interesting to investigate with the 3D grid. For the flat nature of public spaces and also as the attractiveness spread is interpreted as a human walk, the 2D square grid is applied (figure 1). Case study is located in a specific territory of the old town in Lisbon which characterize with a binary type of spaces, public or private. No semi-usage spaces are visible which make place homogenous and beneficial for the clear analysis. Blueprints of buildings are translated as the lack of a grid, hence only the public space is populated with the grid (figure 2). This generates a question about rules of cells situated closest to the edge (see “The closest neighbourhood”).

**States of the grid cells**

Conway’s Game of Life was characterized with two types of cells, a dead and a life cell (Gardner 1970). This binary concept is adopted into the attractiveness simulation (figure 3). Therefore, there is a distinction between interest, or the lack of the interest which results with white or black colour of the cell respectively. Constituting more cell states could give stochastic results which might differ from the actual, more intuitive results.

**Initial state**

As both, SG and CA start from the initial shape, relevant data had to be analysed to construct the initial state of the design. The initial life is defined by the actual location of amenities (figure 4). Amenities meaning facilities which make the place more attractive to people. To name few: restaurants, museums, shops, pubs, benches, monuments etc. It is quantity, not quality taken under consideration. There is no difference between one museum, or one bench location. The data is obtained from the open source platform Open Street Maps (OSM) [1] in an “osm” format file which is translated into a set of geometry in

The square grid is overlaying the buildings’ outlines map. The size of a cell is 400cm by 400cm (figure 4) and is interpreted as a step. The size of a cell is adjusted to fit at least 3 cells in a row or column in the narrowest area. Thanks to that at least one cell has a full Moore’s neighbourhood on a certain level, except for the end of the street and another specific situations. Vertices of the grid, excluding the ones inside the buildings’ outlines, are the cell point references. As amenity location is not necessarily in the exact location of the grid vertices, the closest vertex to the amenity location becomes the initial life cell. The rest is dead. This is how the initial shape is produced.

The closest neighbourhood

People are multi factor decision makers. The closest neighbourhood allows to specify the complexity of region to be analysed during the local iteration, see “Iterative Process”. Few important concepts were considered for implementation. J.V. Neumann specified the closest neighbourhood in 2D square lattice as 4 the closest cells (Gardner 1970) (figure 5) which does not allow the diagonal cells to be analyzed. E. Moore defined it as 8 closest cells [6] (figure 5) which allow the diagonal cells to be analysed. Another complex model assumed cells in a radius of 8 cells to be the closest neighbourhood which result with 200 neighbours (White and Engelen 2000). In this case, certain hierarchy of cells values is necessary to develop. Also, non regular grid systems with cadastral neighbours are noted (Stevens and Dragičević 2007) but it does not reflect the nature of human walk as is to be analyzed. Authors decided to apply rules on the preference of 8 closest neighbours, which is the Moore’s definition.

Since the grid is irregular there are two main narratives of cells which are the closest to the border. As transition rules base on amount of life cells, border cells will have different chance to become life than cells with full closest neighbourhood. Firstly, border cells can be ignored during iterative process and stay frozen in the initial state but still interpretable by the neighbour cells. This could affect the final output. Authors assumed that in pedestrian walkways people would prefer middle of a street rather than walking close to a wall. In the interest of that, border cells are treated with equal rules as the others.

Transition rules

This is a hearth of the Cellular Automaton. Self-organizing behaviour relies on transition rules and executes each local iteration on their basis. To clarify, I will use a “life cell” to name attractive cell and a “dead cell” to name an unattractive cell.
All rules are based on amount of life cells in the Moore's neighbourhood. Each rule is illustrated with two states of 9 cells - before and after the rule application. Thanks to the compound grammar (see section “Iterative process”), the state of analysed cell is described either by number 0 (dead cell) or 1 (life cell).

Rules are divided into four types (figure 2):

1. Analyse the dead cell and transform to life cell
2. analyse the life cell and stay life cell
3. analyse the life cell and transform to dead cell
4. analyse the dead cell and stay dead cell

The first type (figure 6) brings the life from dead. This rule type is applied when 2 or 7 cells are life neighbours. From the urban perspective, it is the optimal location for new services. The area can be re-thinked in terms of public transport, need for services or the quality of street lights. Psychologically, the environment is interesting and balanced. A need for new life is observed and the market has to be fed.

The second type (figure 6) states the cells are still life after the local iteration. This rule is applied when amount of the life neighbours is 3, 4 or 5. Literally, it shows a balanced environment so the surrounding conditions allow the life stay in place. From the urban perspective the amenities are well located to produce an attractive space. Pedestrians are not bored nor overwhelmed with the attractiveness of the surrounding.

In the third type (figure 6), rules transform a life cell into a dead after the iteration. This occurs while the amount of the life neighbours is 0, 1, 2, 6, 7 or 8. This is the case of either insufficient amount of activity in the neighbourhood to sustain life or the overpopulation case. Practically speaking, the amount of neighbouring attractive cells is too low to maintain life. The area is imbalanced and the attractiveness rate does not allow for a new life generation.

The overpopulation case seems arguable, as the higher the attractiveness, the better for the growth of interest. Albeit not. Growing amount of people appreciate space comfort. This approach states
about overpopulation case, which is meant to reduce amount of life in the case of over attractive space. Mona Lisa by L. da Vinci is an unarguable masterpiece with extreme rate of attractiveness. Everybody wants to see the painting but the crowd is inevitable which destroys the sensation. Similarly, compare an exclusive restaurant in a well-known place, with a long queue pending and a restaurant with a shorter queue, but less known. Statistically, we would enjoy the meal more in the latter one.

The fourth rule type (figure 6) remain the cell dead. This occurs while the number of the life neighbours is 0, 1, 3, 4, 5, 6 or 8. In effect there is no life appearing which means that the space is not attractive and did not use to be. This does not mean that after the next iteration the cell will be still dead since each iteration usually changes the neighbourhood. Large amount of life neighbours seem to attract life but when rules change to the lower amount of surrounding life, the map is rapidly populated with life cells.

The number of life neighbour cells need to decide on the cell transformation is selected after multiple experiments with the rules. Sometimes the rules resulted with total life aggregation in few iterations or a complete life extinction. With the above rules a certain balance and intuitive behaviour is achieved. Worth of elaboration is the algorithmic optimisation of the rules to achieve certain predicted, or measured evolution.

A sequence of discrete time step
The interval time for each iteration as a real time interval can be measured with setting the initial state from the known state of amenities in the particular year and to compare it with the following years. Monitoring the amenities growth and calculating how many iterations lasted the simulation to develop desired conditions could give an approximate result on the time interval of one iteration.

ITERATIVE PROCESS
Two SG algebras are applied to distinguish local iterations from the global iteration (figure 7). A local iteration analyse the closest neighbourhood only. The process of local iterations is finished while all cells are analyzed. Subsequently, the 0 and 1 values from local iterations are translated into white (life) or black (dead) cells. This procedure is called a global iteration. Having done a 100 iterations, we have in mind global iterations. This compound grammar consists of algebra of labeled shapes (white and black cells) and algebra of sets of labels (numbers)(figure 7). The first algebra does not influence the latter. For that reason a algebras translation rules has been developed (figure 7). As two following local iteration regions can intersect in 4/9, this would dramatically false the result. One local iteration does not disturb the next one. A local iteration analyses cells and outputs a number as a result (figure 7). Without compound grammar SG would give arbitrary results which are unacceptable. In the Python scripting language there is no such an issue because the procedural code executes the analysis of cells first, then applies the new state of cells.

CA IN PYTHON FOR GRASSHOPPER
The authors decided on using the Python scripting language in Grasshopper to process the rules in an efficient way. The pseudo code (figure 8) consists of input data, processing components and output data. Buildings outline, the amenities location and the region of analysis, they define and construct the grid. Cell size must be specified to define the scale of simulation. The grid preparation is designed with Grasshopper components and “Elk” Grasshopper plug-in [4] to extract the data from the open source maps (see “Initial state”). The grid preparation step outputs Moore's neighbourhood point in-
dexes and the boolean state of each cell. The boolean “True” means the cell is alive, the boolean “False” means the cell is dead. The data is passed to the “CA engine” which executes two more inputs. Iterations count (99 in the case study) and the CA rules to be processed during each iteration. Rules are the same as described with SG (figure 6), but formulated in a conditional way. If the condition is satisfied, a particular transition is executed. The CA engine generates two outputs, new boolean state of cells is used to visualize black and white cells. The frequency of life of a cell indicates the proportion of the iterations a particular cell was alive out of all the iterations carried out. This data is processed into a visual representation section (figure 10) and the frequency graphs (figure 11 and 12).

RESULTS
A hundred iterations are executed to specify the final result (figure 9). The analysed area is 212m per 212m. The 400cm size square cells are set in the 53 per 53 cells grid. It gives 2809 cells, after mapping them on the public space 2395 cells are left to be analysed. In total 239 500 local iterations are being analysed to give an interpretable result.

First the output was registered with dead (black) and life (white) cells only (figure 9). Four general behaviours are visible. Behaviour A, the rapid population. In the initial state the region is dead. After few iterations intensive life cells generation is observed (figure 9). This case occurs when the narrow, populated with dead cells, initial street is in the proximity of the initial life from both sides. This case is observed 3 times. Behaviour B (figure 9), extinction of the life cells from the region. Happened in the initially populated with life narrow street, resulted with extinction of life after 70 iterations. Behaviour C (figure 9), connection of life from two separate life islands. Observed between the two concentrations of life cells in the wide street. Behaviour D (figure 9), spontaneous generation of life. In the 100th generation, initially...
empty region is surprisingly inhabited with the life cells.

This iterative analysis shows the actual state of cells in the particular iteration. No record of the previous generations has been registered (figure 9). The life frequency analysis focus on summing the quantity of life cell state during 100 iterations (figure 10). The more reddish the colour, the bigger count of life cells during the iterative process.

This method contributes to unification of the result and getting rid of extremes with short lasting and rapid changes. Each of 100 iterations have a record and affect the result. To compare the previous, black and white analysis, the same regions are taken for the closer look. In the region A, the life populated corridor was maintained long which resulted with the reddish colours (figure 10). In the region B, both simulations present similar evolution - extinction case. In the region C, the connection of the two life islands happened similarly to previous simulation. In the region D, the difference is visible (figure 10). The spontaneous life outburst in the black and white simulation, was not recorded during the life frequency analysis. This reasoned with low frequency of the life appearance in the region.

The attractiveness graphs are prepared to relate count of attractiveness with the particular cell during single iteration (figure 11). Order of cell location goes with columns top down from the left top corner on the simulation graphic. They depict the tendency for
growth of some regions from the beginning presenting certain regularity in amount of the life cells. The two summits show the most attractive space which is visible as red regions in the frequency of life cells analysis (figure 10).

The attractiveness topography graph (figure 12) presents 100 iterations of 2D frequency of cells attractiveness graphs (figure 11). This allows to observe the tendency of evolution upon the time increment which is specified by the iteration count. Graph summits tend from the very beginning to be the most attractive at the end. Graph valleys show less attractive corridors.

**FUTURE ENHANCEMENTS AND CONCLUSION**

Experimenting with the shape of grid is worth of a closer look. For an instance, a hexagonal grid results with the same distances from point to all neighbouring cells. This would reduce the heterogeneity of cell distances which is \( x \) or \( x\sqrt{2} \) (figure 5). Plenty of rule configurations have been tested and the most intuitive to authors were selected. Worthy of development would be the algorithmic optimization of the rules to achieve beforehand known state, this would give another reason for the prediction legibility. To boost predictability rate, monitoring of amenities distribution growth would be necessary. Comparing real world results with the algorithm results would not only measure the predictability rate, but the time interval of iteration. Rules for implementing new amenities would be interesting to design. This would clearly demonstrate the most attractive place in the analysed region and would influence the further growth.

The paper presents the first approach to prediction of the space’s attractiveness with generative, self-organizing tools such as Cellular Automata and Shape Grammars. The two models complement each other, the SG is used mainly as explanatory grammar and the CA as the performance engine. Attempts to define rules resulted with maps showing rule-based topography of attractiveness over an undefined time interval.

The combination of two shape concepts (SG+CA) illustrates the importance of shape generating simulation to be used in the early stages for design decisions. Having in mind the importance of shape to convey complexity and to get a fast reading of it.

**REFERENCES**


Speller, TH, Whitney, D and Crawley, E 2007, ‘Using shape grammar to derive cellular automata rule patterns,’ *COMPLEX SYSTEMS-CHAMPAIGN*, 17(1/2), p. 79


Stiny, G 1980, ‘Introduction to shape and shape grammars,’ *Environment and planning B: planning and design*, 7(3), pp. 343-351


White, R and Engelen, G 2000, ‘High-resolution integrated modelling of the spatial dynamics of urban and regional systems,’ *Computers, environment and urban systems*, 24(5), pp. 383-400

[1] https://www.openstreetmap.org/