AGENT-BASED MODELS FOR COMPUTING CIRCULATION

Renee Puusepp
Estonian Academy of Arts

ABSTRACT
This paper investigates conceptual modeling of circulation in settlements using agent-based systems. It explains and compares two different types of model that highlight diametrically opposed architectural tasks in respect to movement. While the first model explores the case where the agents’ movement defines the circulation, the latter builds towards the argument that good circulation can be achieved via self-organization within the agent colony. The first type of computational model generates networks that are directly informed by the acts of movement and circulation is a direct result of agents interacting with their environment. Circulation in the second model is seen as an integral part of the morphogenesis of the settlement geometry. The two models help drawing the distinction between two different notions of agency in design. On one hand agency is seen as a group of actors that organizes and modifies the design solutions or its representations. On the other hand, the configuration and structure of agency becomes a design solution per se.
THE ESSENCE OF CIRCULATION

Circulation is seen as an area dedicated to movement and connecting various parts of a building or settlement together into a coherent network of spaces. Crossing borders between private and public spaces, circulation provides opportunity for pedestrian and vehicular access. The circulation network creates and handles flows, links activities together and produces the continuity of space. The circulation network channels matter and energy between spaces—enables and at the same time is informed by the acts of movement.

Movement can be a major shaper of the environment. As opposed to many other issues in design, the circulation lends itself quite easily to computation. With respect to mobility, Rudolf Arnheim points out that there are two basic solutions available for building design—the shelter and the burrow (Arnheim 1977). Whereas the abstract type of burrow is the result of inhabitant’s physical penetration through the environment, a shelter cares about its user’s movement only secondarily, and derives its form from its own function instead. The burrow type of buildings is directly informed by the user’s acts of movement, providing more space at locations where the user wants more freedom of direction (Arnheim 1977).

The burrow analogy becomes useful when building computer models for generating design typologies where circulation constitutes the major part of the venue. Airports, racing courses, subway and train station all feature large dedicated spaces solely intended to accommodate movement. Ancillary spaces in these venues generally follow the circulation logic. In hospitals and schools, the circulation network is also a key driver of spatial organization.

The circulation network has to handle anticipated flows and provide access to desired spaces. An obvious aspect is also the economy of the layout—solutions that optimize the length of journeys between connected spaces are usually better in terms of occupational and construction costs. The quality of circulation in such cases can often be assessed by quantitative means, making the computational modeling suitable for generating the burrow type.

However, circulation in the shelter type of buildings and in most urban environments cannot be solely designed to the optimal network length. Kevin Lynch (1981), for example, affirms that circulation is not just about shortest paths and can provide aesthetic pleasure, hence the optimization of the road and corridor lengths often comes at a cost. Circulation in these cases is clearly an important and integral part but not necessarily the key driver of the design; it has to reiteratively co-evolve with the rest of the solution.

ALGORITHMIC SOLUTIONS—A BRIEF HISTORY

Algorithmic solutions to spatial problems have generated enthusiasm in many architectural theorists and fascinated practitioners for decades. With the ever-increasing availability of computing power, more and more of computational design solutions can be and actually are tested in practice. Generative models shed light to network formation processes, offering an alternative view to descriptive models. Descriptive steady-state models are not helpful to understand dynamic networks since the networks are usually formed by an iterative step-by-step process (Haggett and Chorley 1969). Lynch (1981) adds that the computer makes it possible to explore a view of the city different to an intuitive and descriptive analysis. The computer can help us model the city as “the cumulative product of the repeated decisions of many persons and agencies—actors who have diverse goals and resources” (Lynch 1981, 336). Lynch seems to believe that when actors in such models are made to represent pedestrian activity, it is not difficult to see how the model becomes a useful tool for solving circulation issues.

The tractable and quantifiable nature of networks has resulted in abundance of computational models for generating circulation systems. The vast number of solutions that have been proposed for solving minimal spanning tree and traveling salesman-type problems is beyond the interest of this paper. However, it is worth mentioning Adamatzky’s (2001) approach for solving minimal route problems. He has shown that route planning can be computed in simple cellular automata collectives.

The computational models for finding suitable circulation layouts are often accompanied by automatic activity location procedures. The extensive exploration of heuristic models for automatic floor plan and circulation generation in 1960s has been documented by Tabor (1971)—several computational modelers in that period have engineered computational methodology for factories, hospitals, educational buildings and offices. All these models broadly represent the mechanistic view on circulation and tend to leave out other spatial qualities. The generated results are usually diagrams that are intended to satisfy the circulation condition and do not take into account organizational, functional, environmental, geographical, structural, legal, or financial criteria.

Somewhat more advanced are the contemporary office layout generators that drive the floor plate shape from circulation (Shpuza and Peponis 2006). To guarantee that the final design matches given criteria, Shpuza (2006) chooses a preferred circulation system in advance, but also explores the potential of deriving circulation from the floor plate shape.
Several shape grammar-based circulation generators have been developed by various authors. For example, Pascal Mueller has participated in creating a procedural city modeling software called CityEngine that enables designers to quickly grow street networks. The generative method behind CityEngine is a shape grammar similar to context sensitive L-systems (Chen et al. 2008).

Other generative models benefit from well-known methods and processes in evolutionary computing, Artificial Intelligence research and related fields. Zhang and Armstrong (2005) have introduced genetic algorithms to locate corridors in a 2D lattice of cells. Diffusion limited aggregation has been used to explain growth processes of city networks (Batty 2005), or to generate street networks (Nickerson 2008). A computational model for generating leaf venation patterns (Runions et al. 2005) has produced circulation networks that share many commonalities with street networks and can be considered for architectural and urban design purposes.

Few agent-based models have also been used for generating circulation systems. Jones (2010) has developed a computational model where the mobile agent colony creates dynamic transport networks in response to a chemical stimulus. Goldstone and Roberts (2006) have proposed an agent-based model to study and reproduce self-organized trail systems in groups of humans. Ireland (2008) has introduced an agent-based model that also tries to sort out the desired relationships between activities in a building. An agent-based model simulating the growth of networks is inspired by biological tunneling ants (Buhl et al. 2006).

Despite the number of computational examples, all the investigated circulation modeling methods are either mechanistic, reductionist, tend to focus on network optimization, seldom produce solutions that are relevant to architecture or city planning, or are only good for analyzing the network formation phenomena and are not tested as design tools.

The following two sections introduce agent-based circulation modeling methods that are simple yet flexible enough to be plugged into the urban design process. The first model could serve as a design tool for city planners wanting to quickly generate and compare different options of street networks. The second one is an integrated solution where circulation is developed as an emergent by-product of global morphogenesis of the built form. The latter one suggests that circulation network does not necessarily have to be the direct result of agents’ movement. Instead, circulation can be formed of agents.

**EMERGENT PATH FORMATION: CONSTRUCTIVE DIAGRAM GENERATOR**

The Constructive Diagram Generator is conceptually very simple prototype. Yet, for its relative simplicity, it is capable of producing a significant variety of movement patterns (Figure 1). The algorithm that drives the prototype is inspired by the path formation behavior found in social insects, particularly in termites (Bonabeau et al. 1998) and ants (Deneuborg, Pasteels, and Verhaeghe 1983). The use of software agents to mimic the behavior of natural insect colonies is an obvious choice: if one can simulate the path-finding behavior of insects, one can also reproduce the path formation in silico.

Unlike ants and termites, the agents in the Constructive Diagram Generator prototype are not driven by the need to survive, but are simply locked into a simple sensory-motor loop. These agents move around in the environment and are drawn to certain attractors or markers,
computational representation of pheromone. By moving around, they alter the attractiveness of particular locations that, in turn, attract more agents. As the result of such a simple feedback mechanism, a colony of agents can produce looping movement networks–macro patterns that emerge from micro behaviors. To put it differently: agents start following their own trails, and thus generate continuous movement flows. The generated patterns manifest the continuity of agents’ movement and are facilitated and constrained by their maneuverability and sensory configurations. The Constructive Diagram Generator is by no means intended to be a plausible model of natural phenomena; the objective is to investigate a simple and robust prototype for generating circulation that can inform, drive and be integrated with other design computing methods. While the model has been extensively explored to study the behavior of social organisms, its capabilities as a design tool have remained relatively unexplored.

In its core, the model is fairly similar to the Jones’ computational model that generates spontaneous transport networks by mimicking the behavior of the true slime old *Physarum polycephalum* (Jones 2010). Jones explains how the emergence of movement networks can be based upon requirements in foraging behavior: covering maximal search area with the least energy expenditure. Jones’ agents are fairly simple mobile particles featuring three sensors that help agents identify the chemical gradients in the environment.

Agents not only sense the gradient field but also change it by adding more chemicals locally leaving trails of movement behind them. Their environment is composed of a grid of discreet cells that store the chemical. The chemical evaporates over time, making the trails gradually fade and eventually vanish. Given the right sensory configuration, Jones’ agents leave behind trails that form complex network patterns approximating hexagonal tiling. The dynamic formation of movement network can also be affected by modifying pre-pattern stimuli weights. Jones shows how inserting spots of concentrated chemo-attractants into the environment leads to the emergence of movement network that approximates Steiner minimum trees.

The ability to modify pre-patterns reveals the model’s potential as a design tool. One can easily see how controlling the attractor points opens up the opportunity to use the dynamic network formation process to inform planners and architects at the conceptual design stage. Naturally, the output of the model–trails left behind by agents—need to be further interpreted and perhaps are the best described as constructive diagrams as defined by Christopher Alexander. Alexander (1964) points out that there are two types of diagrams–form and requirement diagrams. Whereas the form diagram refers to a physical shape, the requirement diagram is a non-iconic notation of some constraints or properties (Jormakka 2002). The latter is only interesting if it somehow informs the former. Combined together, these two types result in a diagram that Alexander calls a constructive diagram.

Jones’ model (Jones 2010) serves as a good foundation for developing a design tool – the Constructive Diagram Generator. In addition to the Jones’ methods, the Constructive Diagram Generator allows the designer to define attractor points for agents that create gradient fields of proximity across the model. Agents choose their movement direction by combining information from this static gradient field with information from the dynamic field of chemo attractant. Pre-computed proximity fields automatically give direction to the agents’ movement, making it possible to simplify their sensory configuration. Agents no longer need to have a predefined sensor configuration but can simply retrieve information from adjacent cells around them. Simpler sensory configurations require less computing and theoretically lead to a better performing design tool.
The Constructive Diagram Generator was tested against an international design competition brief in Copenhagen. The competition entries were asked to propose a solution for a large industrial peninsula with several basins and quays. The competition brief envisaged a regenerated area with 40,000 new residents and the same number of new jobs created. One of the major requirements was to design an efficient transportation network prioritizing walking, cycling and public transport over the use of private cars. The Constructive Diagram Generator allowed creating a variety of diagrams that meet the efficiency requirement but also allow exploring different options to locate main attractor points in the area such as schools, shopping and entertainment centers (Figure 2). Smaller attractor points were uniformly distributed across the entire area ensuring access to residential and office blocks and providing sufficient permeability and connectivity in the street network. The designer was thus able to control the model in two ways: preparing the boundary of the simulation and locating the target nodes within the area.

TOPOLOGICAL SETTLEMENT STRUCTURES: DWELLING AGGLOMERATOR

Most of the existing agent-based computational models including the Constructive Diagram Generator create circulation networks as a result of agents’ movement and interaction with their environment. Circulation networks in such models are generated in isolation from the morphogenesis of the settlement geometry. Thus, circulation becomes either the prerequisite and the generator or a derivate of the non-circulatory spatial structure.

However, there is an alternative to this approach – circulation can be generated as a by- or sub-product of overall morphogenesis of the settlement geometry. In such cases, circulation is not a discreet network as known in graph theory, but as space that facilitates good access to all built up areas. This paper argues that by incorporating right spatial analysis methods, settlement layouts with high level of access and high permeability can be generated.
The concept has been invented and tested already in 1980s by Bill Hillier and Paul Coates. Their research shows how the built form and circulation in unplanned settlements emerges solely out of local interactions. Hillier illustrates it with an extremely simple yet conceptually powerful model – the Beady Ring model (Hillier 1989). Space is treated as if it was composed of nothing but mobile individuals. Hillier explains that in such a discrete system, individuals can give rise to the settlement structure by simply reacting to local rules (Hillier and Hanson 1984). The Beady Ring model – or Alpha Syntax as Paul Coates (2010) calls it – relies on a simple principle of positioning and orienting dwelling units in a settlement so that no units share a common entrance area with any other unit in the settlement. As the result, the settlement features a highly diverse yet well-connected street network. The Dwelling Agglomerator prototype has been directly inspired by the Beady Ring model. Unlike its precursor, the Dwelling Agglomerator has a potential of becoming a design tool. Its modular nature makes it extendable and compatible with standard design analysis routines such as environmental and structural analysis. For the purpose of this paper, the prototype is only described from the circulation and accessibility perspective.

The Dwelling Agglomerator is composed of three programmable modules and two generative cycles–self-organization and the larger adaptive development cycle (Figure 3). The first module is concerned with the self-organization of individual dwelling units and relies on an original movement algorithm. The global position of dwellings and their relationships with the neighboring units are dynamically determined during the reiterative process within the module. The second module features a set of parametrical rules for modeling the agglomerate geometry by connecting the neighboring units together into clusters of dwellings. The third module performs an analysis on the created geometry and assigns a grade value to each dwelling unit based on accessibility calculations. Whereas all three modules are connected into a repeatable programmatic loop, each module can be also edited separately.
The second part of the proposed model takes care of the parametric shape manipulation. It suffices here to say that the shape calculation is based on the Voronoi spatial subdivision method. The third module performs quantitative analysis on the generated geometry, calculates which units are ‘satisfied’ and can thus maintain their position, and which units should find a new and better place. This information is then sent back to the very first module. The grade values assigned by analysis determine how the movement on individuals is adjusted in the next loop – ‘unsatisfied’ individuals with low-grade values become active in finding new locations.

Satisfaction of individual agents can be calculated depending on the design task. For the purpose of this paper, the generated layout is analyzed in terms of accessibility. In order to perform the accessibility calculations, the space between dwelling units is also modeled as a dynamic agent colony of discrete spatial units capable of changing their location. But unlike dwelling units, these agents are purely reactive, filling in the spaces between dwellings and adapting to the changing configuration of the settlement structure. Consequently, the entire model is one large multi-agent system composed of active dwelling units and ‘empty’ spatial units representing the public domain – the circulation units.

Accessibility of each unit is defined as a sum of proximity values to all dwelling units. The value is inversely proportional to the topological distance between the unit and dwelling units. Topological distance is effectively the smallest number of steps that it takes to reach dwelling units from the given unit via circulation units. Dwelling units cannot be used for circulation and as a result, the units that are surrounded by dwellings have lower accessibility. However, low accessibility does not cause units to change their location. Instead, dwelling units that reside next to a low value neighbor are identified as blockers and are forced to move away from their current location thus creating new access routes to the unit and consequently increase the unit’s accessibility.

(Figure 6) depicts the sequence where the highest contrast in global accessibility values between neighbors causes the ‘blocker’ unit to move away and create a new access route. ‘Empty’ space units (white cells) fill in the gap.

The self-organizational process (Figure 7) of the settlement structure is thus driven by two competing algorithms – the flocking of dwelling units and the access blocker identification routine. Flocking ensures that the dwellings form coherent clusters while the latter algorithm breaks down the barriers to enable fluent circulation in the settlement (Figure 8).
Circulation units are relatively passive compared to the actively flocking dwelling units. Although the reported research has not demonstrated how circulation units can actively participate in the morphogenetic process, it is possible to imagine how circulation units have their own flocking rules that help achieve desirable shape for open circulation space. Therefore, circulation does not need to solely react to dwellings but can be a subject of its own satisfaction criteria too.
The modular nature of the proposed computational model makes it possible to combine other kinds of analysis with accessibility calculations. The flexibility of the proposed method lies in separating the satisfaction analysis from the movement behavior. The analysis module always operates on the same input format – simple mesh or polyline geometry – and produces the same output format – integer values of ‘satisfaction’ assigned to each unit. The consistency of input and output format allows stacking up the analytical tests in order to take multiple design criteria into account. (Figure 9) depicts a result where all dwelling units are well accessible and also receive enough direct sunlight.

CONCLUSION

The paper introduced two bottom-up models for computing circulation in settlements. The two models outline different ways of conceptualizing the notion of agent and agency. In the first model, agents are inhabitants of the model – they constitute an agency interacting with its environment. The environment serves as an input for agents’ sensory system and consumes their motor output, but is a completely separate system from the agency. Agents in the second model are quite a bit different – they actually constitute the environment. This underpins a new concept of space as not simply an environment to mobile agents but as an active substance entirely made of embodied actors. Perhaps the biggest contribution of the proposed model to the fields of architecture and urban design is this divergence from the dualistic division of space into objects and void. Instead, space truly becomes an agency, capable of reconfiguring itself and producing design solutions via the process of self-organization. It is difficult to judge whether the concept of space as an agency has larger implications beyond the design field, but it certainly helps conceiving novel types of computational design tools.

The bottom-up approach to designing circulation has two clear advantages over traditional top-down methods. Firstly, the design processes can be extensively automated and require little if any human intervention. Secondly, parallel processes in the agent-based simulation lead to more ‘democratic’ distribution of spatial qualities (for example sufficient sunlight and accessibility) across the layout. As long as the bottom-up algorithms are seamlessly integrated with standard CAD packages, it is easier to accomplish the uniform distribution than it is with top-down methods.
There is a question whether the bottom-up methods would generate settlement layouts that achieve the necessary collective effects, such as good overall permeability and connectivity in parallel to the satisfaction of local criteria. To date, there are no known models that answer this question. The modeling concept proposed in this paper offers a possible answer: conceiving space of discreet agents allows integrating several generative design cycles seamlessly into a single loop and developing solutions that solve more than one design task in parallel. Even though this has only been tested in the abstract settlement model, where clusters of dwellings are formed together with the emergence of street network, successful trials suggest that it is possible to build integrated bottom-up computational models, where multiple design goals are sought simultaneously.

REFERENCES


Runions, Adam, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaelle Rolland-Lagan, and Przemyslaw Prusinkiewicz. 2005. “Modeling and Visualization of Leaf Venation Patterns.” In SIGGRAPH.


IMAGE CREDITS

All image credits to Renee Puusepp (2014).

RENEE PUUSEPP is a practitioner and academic of computational architecture. He runs a London-based technology company SliderStudio and works as a researcher at the Estonian Academy of Arts. Renee finished his formal training and started professional career as an architect in Estonia, then completed a MSc in Computing and Design as well as a PhD in Architecture at the University of East London. In 2011, he was awarded a doctorate degree for the thesis investigating agent-based techniques for advanced architectural modeling. His special interests lie in using generative, parametric and programmable tools at the early stages of design process.