Agent based design systems could provide useful decision help for architects working on spatial planning tasks that involve large number of actors or deal with complex urban situations. These systems are especially helpful in bridging the gap between concrete design proposals and high-level design abstractions such as frequency and flow diagrams. Every attempt to use computational design agents in the planning process will automatically raise many fundamental issues about spatial perception and representation of the environment. The paper discusses these issues in the light of some recent agent based simulations. Two case studies are presented in order to demonstrate different uses of computational agents in urban design. The first study shows how a simple agent-based design system placed in an urban context becomes a creative production tool. The second one reveals analytical capabilities of an agent system in urban environments.
I. Introduction

The aim of this paper is to show potential uses of computational agents for designers. Agent modelling techniques offer an alternative approach to the common practice in urban and architectural design. The very distinction agent systems possess is situatedness; agents are directly embedded in the simulated environment. To illustrate the opportunities of situated agent models for architecture, the authors of this paper have selected two case studies. The first study is an application of agent-based design system that simulates flows of urban actors, whereas the second one is concerned with cognitive aspects of simulated space. These two studies have been presented in order to draw the line between generative and analytical methods in computational urban design. The generative model deploys design agents in an existing urban environment. The analytical one shows how a computational mechanism could identify urban layouts that impede wayfinding processes.

Computational tools for generating architectural solutions require a modeller to clarify quite a few fundamental issues about space. This paper attempts to address some of the issues regarding spatial representations and proposes a set of abstracted computational models. The objective of simulation models is not to replace the human designer but rather alter traditional design methods and provide bottom-up analytical and generative contribution to the design process.

Simulation models tend to focus on specific properties or behavioural patterns and explicitly consider some components of the explored environment while ignoring others. Therefore models always exist in vacuum, in an artificial isolation, and remain abstractions of the phenomenon they are mimicking. Which components are engaged in a model, is entirely up to the designer of this model. With the growing degree of abstraction the designer loses the adequacy of the model that could then just become a system per se.

One of the challenges the designer is faced with during modelling an agent based simulation, is to define relationships between agents and the simulated space. Luhmann’s [1] description of system differentiation becomes a useful concept for a simulation designer. This concept helps the designer to draw boundaries between different systems. In Luhmann’s terms everything that is external to an agent (as a system), including other agents, becomes the environment. The question of what the environment is has to be replaced with the question of what it affords (Gibson [2]). The designer only has to specify the connections between an agent and its environment. The systems’ differentiation enables the designer to disregard the notion of space and integrate it seamlessly into computational models. From Luhmann’s system theoretical point of view, space is a part of the environment that does not need a separate definition.

Two case studies presented in this paper display different approaches to computational agent simulations. The first study is a part of a concept design...
level project commissioned by an architectural practice (Aedas, Edinburgh office), and is developed in collaboration with Christian Derix. It uses an agent modelling technique in a top-down generative way but still exhibits some emergent properties of distributed systems. A reaction-diffusion model described by Adamatzky [3] is used to map simulation data to spatial representation. Due to the nature of the project there are no cognitive powers assigned to the agents. In contrast to the first study, the second one takes place in cognitive space [4] rather than in metric space. The study employs a notion of cognitive mapping and tries to reveal some computable aspects of it. The cognitive map is seen as a network of values that agents can read and change. During the exploration in a CAD model, agents develop a set of formal rules to interact with and move around in the digital model.

2. Related work

2.1. Emergent patterns

There has been a remarkable growth in using distributed simulation models in the past decade. Agent based simulations have found its place in robotics; have been implemented in game design, and social, environmental and complex system studies. These models vary in their purpose and implementation technique defining the concept of agent in many different ways. In most cases, agents are seen as units that have a certain degree of freedom in their movements or are autonomous in their other behavioural decisions, and are distinguished from their environment.

Resnick, the creator of StarLogo, introduces the concept of freely moving units (turtles) that inhibit an environment made of static cellular units (patches) [5]. This is a topological distinction that helps to classify an agent, but does not limit the concept of it. In cellular automaton models, such as Game of Life, the concept of mobile agent emerges from the topologically fixed environment. The impression of movement is generated by switching the state of neighbouring patches. Butler et al. [6] introduce a similar model that uses a cellular automaton to construct an agent that is capable of exploring unknown environments and surpass obstacles in it.

During past years, the authors of this paper have discovered that, regardless to the design of the distributed system, similar spatial patterns

![Figure 1: Formation of voronoi-like spatial subdivision using mobile agents](image-url)
can emerge. Namely, similar voronoi-like structures have been achieved with fixed topology cellular automaton models and freely moving agent models. The cellular automaton model uses a reaction-diffusion algorithm; the agent model uses an attraction-repulsion algorithm. The reaction-diffusion algorithm works on a rectangular or hexagonal lattice of cells of which each cell has two possible states, excited or dormant. When a number of cells are activated (source points) the excitement is propagated over the lattice from a cell to its neighbours using von Neumann neighbourhood function. If an inactive cell is simultaneously activated by three or two (at the edge of the lattice) neighbours, the model has found a point that is roughly within the same distance from the initially activated source points. A similar emergent pattern has also been produced with mobile agents deploying the attraction-repulsion algorithm. The tested simulation (Figure 1) had two species of agents – the ‘reds’ and the ‘blacks’. The blacks, simply trying to maintain the maximal distance from the closest red, form the edges of polygonal areas; the pattern that emerges is a result of local interactions between individual agents.

Some contemporary urban theories see the city as a network of patches where free agents represent individuals, groups of individuals or other social institutions (e.g. Portugali [7]). Shane [8] also mentions patches in several occasion, referring to the cellular nature of the urban environment. He uses the term ‘urban actor’ to describe the shapers of the environment without explicitly specifying what or who an actor could be. In conceptual modelling for architecture an actor is perhaps a concept similar to the concept of agent in computational simulation models.

2.2. Learning agents

Agent modelling has been widely used in conjunction with different learning methods. As Bakker and Schmidhuber [9] point out, agents are often difficult to program, and through learning the agents could adapt their behaviour to environmental complexity. The mentioned authors use a type of reinforcement learning. According to Pfeifer and Scheier [10], reinforcement learning is, alongside with unsupervised learning and in contrast to supervised learning, a biologically plausible method. Artificial neural networks are probably the most exploited learning devices that are preferred due their speed, and capability to coordinate sensory data with motor output, supporting all the previously mentioned learning methods. As Stanley et al. demonstrate in designing NERO video game [11], neural nets give agents the flexibility to adapt to changed environmental conditions and develop new behavioural strategies. Stanley and Miikulainen [12] also introduce an evolutionary algorithm that is applied to an open ended learning problem. They use evolutionary techniques setting their agents in competition and triggering coevolutionary ‘arms race’. The mentioned authors also employ genetic algorithms to search an undefined solution space for best control network.
architectures. Besides developing control architectures for agents, computation has also been used to evolve agents with different bodily morphologies. Sims [13] has evolved fascinating creatures in physically simulated three-dimensional worlds, taking an advantage of competition and coevolution, and showing that morphological variance could result in increased fitness of an agent.

2.3. Environment

In a computer simulation the environment is a set of data, an abstraction of the existing environment. Space syntax research has established methods of representing and quantifying spatial configurations by measuring the space in a specific manner (e.g. Penn [14]). This kind of spatial analysis takes into account certain aspects of the geometrical properties of the environment. Turner and Penn [4] claim that from 50 to 80% of variance in pedestrian flows can be explained in terms of spatial ‘syntax’. They introduce an agent based modelling technique merged with syntax analysis. Agents explore an environment that has been precomputed with a space syntax method called visibility graph analysis. That analysis has been implemented locally, so that the agents can access the information that determines what is visible from a specific point in the space. Turner and Penn find that the best correlation between their agent model and observed human movement can be obtained with random-walk agents, without introducing any goals. Thus the agent’s movement depends on the spatial configuration of a specific location and on the orientation of agent. Following two case studies apply a different model, where agents have been programmed to find goals. The environment is preprocessed only to the extent of CAD model in which the buildings and objects are represented by surfaces.

3. Case studies

3.1. Design agents

In this case study an agent system was used to simulate particle flows in order to generate topologically variable spatial configurations. These configurations suggested possible ways of organising pedestrian movement and could be the subject of further computational analysis. However, in current case, the analysis of outcome was carried out by an architect and is not described in this paper. The model was not designed to simulate real pedestrian movement, but generates possible conceptual schemes (Figure 2).
The case study is an implementation of agent-based system in an existing urban situation. The complexity of the site is manifested in large number of urban actors and different system flows that cross the site. The model solely simulates the physical movement of particles between several emitters and targets. Agents do not possess any mechanisms for recording nor organising environmental stimuli; they are not able to learn. Emitters and targets have been set by the existing situation or are controlled in top-down manner to indicate premade decisions. Although the agent-based flow-diagram is fairly simplistic, it produces results that could be unpredictable without the simulation. It is argued that relatively complex outcomes emerge as a result of system-environment interactions, although the system itself is not intricate (Figure 3).

The study is focused on a transportation interchange scheme and explores possible pedestrian routes between a railway station, bus stops and a tram stop. Besides the public transport, the site accommodates a busy crossing. Pedestrian flows are characterised by strong fluctuations and dictated by disrupted inflows. Inflow rates are defined by public transportation itineraries and also reflect the rhythm of city life, growing high during daily rush-hours. The simulation model allows controlling the position of inflow points (emitters) as well as inflow rates, thus enabling one to model various scenarios.

Simulation
To set up the flow and frequency simulation one needs to define flow emitters such as entrances to the train station, bus stops, tram stop and inflows from main pedestrian routes. Definition of an emitter requires selecting a position for it in CAD model of the site, and assigning inflow rate to it. The flow rate specifies frequency and amount of emittance, which corresponds to the assumed frequency of flows on the site. Each emitted particle is an abstraction of an urban actor that strives towards a designated goal and is addressed as an agent.

The agents’ action is based on few simple rules. An agent, released from an emitter, tries to find a new position that is closer to the target than its current position. The agent searches for a new position by comparing a random set of positions within a certain radius. On its way, the agent detects objects that it cannot permeate. The agent can continue its journey by moving along the edge of an object as long as the new position is closer.
to the target than its previous one. Whereas an individual agent is not capable of overcoming some obstacles, agents as a community have a cooperative power to create ‘bridges’ to cross the roads. These ‘bridges’ are simply the points where agents can ‘jump’ over the roads, and are established only when there is sufficient desire from both side of the road to cross it. Every time an agent encounters a surface and cannot find a new position to move, it leaves a marker behind. Markers are indications of a missing crossing. Eventually, when these marks pile up on both sides of the road, the ‘bridge’ can be formed. Such kind of collaborative power does not appear as a result of direct communication between agents but is mediated through the environment.

Interaction between the simple system and its environment lead to behavioural patterns that are often surprising and can hardly be foretold. In addition to the generated crossings, simulated particle flows could be helpful in making spatial-functional decisions. Analysis of the simulated model could detect density of space usage and potentially good and bad location for different functions. Mapping out interferences between agents and cross-flows suggests keeping some areas open to facilitate the emergence of public activity.

The study employs a technique to map flows to spatial structures. The mapping technique uses a reaction-diffusion algorithm as described earlier in this paper (see section 2.1). This algorithm uses a cellular automaton model to generate Voronoi tessellation that can be used in subdividing given space into smaller areas, which in turn could accommodate a range of different functions or share a common spatial property (Figure 4). A Voronoi cell is initiated in the location where two agents collide. The growth of Voronoi structure shows the potential intensity of pedestrian movement and suggests the areas that should be kept free of obstacles. The size of Voronoi cell in fully developed tessellation indicates whether this location can be used for massing or should be left as an open space.

3.2. Cognitive agents

The objective of the study is to come up with a wayfinding mechanism where agents have a minimum number of stimuli from the environment.
Having a limited set of inputs should allow one to find out spatial features that have the greatest impact on the agents’ decision-making. In order to keep the number of inputs from the environment low, a method for storing previous sensory-motor decisions is proposed. This type of memory has many similarities to the concept of cognitive mapping.

The only stimuli an agent receives directly from the environment is limited to the collision detection of surfaces that represent the urban elements. The major part of perceptual modelling is mediated through a cognitive map, which is seen as an internal representation of the environment for the agent colony. While the map evolves during the simulation, the correct rules to interpret the map develop with it.

The study does not try to explain how different organisms find their way in a particular environment. As declared above, the authors of this paper attempt to discover a simple mechanism for a successful wayfinding process and to prove that some spatial features facilitate or hinder this process.

Cognitive mapping

This study poses a contingent way to approach processes of cognitive mapping using computational methods. The used methods do not attempt to explain the full range of natural cognitive mapping, but suggested models are hoped to provide some helpful insights of how spatial experiences can be recorded and reused. The usefulness of discussed maps is displayed in unraveling wayfinding difficulties within multi-agent environment. Learning to find one’s way around in a new environment is considered here to be a sufficient proof of a developed cognitive map. The proposed algorithms try to exhibit essential parts of cognitive mappings such as collecting, storing, arranging, editing and retrieving environmental data.

Cognitive maps are representations of the environment, acquired through direct or mediated perception. Ungar et al. [15] argue that, although the construction of spatial representations is greatly facilitated by visual experience, other modes of perception can generate cognitive maps as well: different modes just lead to different strategies for coding information. Psychologists and geographers have used the term ‘cognitive map’ for decades in the context of wayfinding and environmental behaviour (e.g. Stea [16]). A similar concept was introduced in architectural discipline by Lynch [17] and his theory about the image of city. Lynch uses the term imageability to describe the property of an environment to facilitate or impede the process of image-making. If one extracts the spatial features of highly “imageable” environment, one can design environments that communicate without the help of written signs.

It is reasonable to argue that different individuals or different species perceive the surroundings in their own particular way and, therefore, the imageability is not an objective property of an environment but also
dependent on cognitive mechanisms of a perceiver. An organism’s ability to learn is strongly related to its position in the evolutional timeline. Higher organisms have less intrinsic behavioural patterns, and depend more on their experience (Gould[18]). Environmental learning involves obtaining appropriate sensory-motor conduct, and storing spatial information for future use.

Kuipers et al. [19] describe cognitive maps as topological representations of interlinked places. Cognitive maps possess a kind of continuity; a certain procedural or algorithmic sequence that one follows in order to get from one point to another. The map is more a routine-like description, a behavioural or sequential description of an environment rather than a structural description of surroundings. From architects’ and city planners’ point of view, it is worth studying the process of cognitive mapping and its major shapers, because it could reveal the principles behind environmental decision-making.

Flow of the algorithm

The algorithm under review benefits from several well known concepts. The goal of the agents is to find a way to an assigned target point while interpreting sensory input and altering their internal representation of the environment (Figure 5). Successful agents are awarded by upgrading their value; the value of a non-successful agent is reduced to minimum.

Agents are using a combination of computational cognitive map and syntactic instructions that determine how that map is interpreted. In other words, syntactic instructions are the key for the cognitive map. An agent is ‘born’ tabula rasa – the evolution of the map and the development of syntactic memory is changed by environmental stimuli. Conception and fade-out of cognitive data is simulated using pheromone trail algorithm – the information restored in the cognitive map is disappears gradually. Syntactic instructions have to develop and change according to this dynamic map. The pursuit of targets is facilitated by trivial vision. If the visibility line between the agent and its target is clear, the agent takes an automatic step towards the goal.

Besides individual learning, the development of an agents’ syntactical instructions also takes place at the phylogenetic level. Evolution of agents is similar to evolution of Braitenberg’s vehicle type 6 (Pfeifer, Scheier [10]). A single agent is chosen for reproduction. However, in the process of reproduction only 75% of syntactical instructions of the agent are transmitted to its offspring. The selection of the parent agent is partly up to chance – the best agent (by value – see above) in a subset of randomly selected individuals gets the honour to be copied.

Design of the agent

The design of the agent is fairly simple. The visible entity of the agent consists of the central ‘body’ and six sensors attached to it to form three
symmetry axes. The consideration behind the hexagonal design was to give agents sufficient liberty of motion retaining the symmetry and thus leaving undefined the front and the back. Sensors are combined into three identical axes which have a major influence over the activation function (Figure 6). Each sensor axis has four possible states: two polarised states (only one sensor active), both sensors active, and both sensors passive. All three sensor axes together yield a sensory space with 64 possible input combinations. Sensor morphology is invariant, which means that the body plan of the agent is not capable of evolving over time.

The output space, or the motor space, is much smaller containing only six possible directions. The agent can take only one step at a time, and cannot combine different directions. Although the motor space is relatively small, the mapping of inputs to outputs results in 384 different combinations. The agent’s behaviour is simply controlled by a list of input-output matches, and is completely undefined when the agent colony comes into being (Figure 7). The agent has to create the controller itself by

![Figure 5: Input-output coupling. Agent obtains input from a digital environment and from the cognitive map (agent’s internal representation of the environment). Output is generated interpreting input according to syntactical rules (the key of the cognitive map).]

![Figure 6: Sensory input. Sensors acquire their value from the environment and from the corresponding location on the cognitive map.]

Spatial Simulations with Cognitive and Design Agents
interacting with the digital environment constantly comparing the input-output mapping (henceforth: syntactic memory) to environmental affordances. The syntactic memory consists of two layers: long-term and short-term memory, where the latter acts as the verification layer or buffer to the long-term memory. In addition to the controller, the agent possesses some persistence in its action — if the sensory input appears to be unfamiliar, the agent continues in the previously chosen direction. This helps the agents to maintain their explorative behaviour.

Environment and its representation

The term 'environment' obtains ambiguous meaning within the context of digital computers. In most cases, digital environments are just representations, and therefore leave some room for human interpretation. Although, for true intelligence to emerge, people in the field of embodied cognitive science strictly insist on using the world as its own model (Brooks [20]). The complexity of cognitive mapping, however, encourages exploitation of virtual worlds.

This paper addresses the digital environment from the agents’ point of view, thus omitting the observer’s perspective. The three-dimensional data ceases to be a representation for agents, and becomes adequate environment for them. Agents, in turn, create a representation of this environment, which can be interpreted as a cognitive map. The map is laid over the environment only for observational considerations, but it can actually be an integral part of the agent.

The cognitive map is a hexagonal network of interconnected nodes. Each node contains one or more values (or vectors). If an agent moves in the environment, corresponding vectors on the map are adjusted according to the success of the agent. A node is just a passive piece of information — agents gain meaningful knowledge by comparing adjacent nodes. Nodes also have a tendency to forget their values gradually causing the information on the cognitive map to fade away.

Some observational notes

The progress in the agents’ behavior and development of the syntactical memory and the cognitive map tends to follow a standard pattern. As the first meandering agent finds a way to its target, all nodes on the cognitive map the agent has stepped on are positively adjusted. Such kind of learning technique can be classified as reinforced learning. When the agent tries to repeat the
same route, it may turn out that instructions in the syntactic memory do not match the modified cognitive map. Thus, new instructions have to be invented to ‘read’ the modifications made by the first agent. If the next agent is capable of finding the target in the slightly changed situation (with positively adjusted nodes), it reinforces the cognitive map, and appropriate change to the syntactic memory has been made to interpret this map.

Certain features of the environment facilitate wayfinding, especially those that suit the agents’ ‘anatomy’ and gestural traits most. It is not always the shortest route that becomes most popular - it is usually the most suitable route for a particular kind of agents (Figure 8). The complexity of an environment can be easily assessed by the time it takes the agent colony to navigate through it. The time consumed is usually proportional to the number of changes in the direction of motion. Some routes are more difficult to learn as they require a sequence of such changes. For example the U-turn was easier for agents to pass through than the S-curve (Figure 9).

A few interesting behavioural phenomena can be pointed out:

1. Agents acquire different techniques to move around in the environment. Some of them try to keep away from environmental obstacles, others, in contrast, develop a ‘wall following’ tactic.

2. Some agents tend to travel to locations where they have clear visual contact with their target, without actually getting closer to the target.

3. If a route to the target has been found, some agents still keep exploring and finding other ways than the established one. The firstly discovered route does not necessarily become the most used one.

4. Unplanned competition between different ‘species’ occasionally takes place. The nature of the pheromone trail algorithm prevents agents using the same trail in both directions. One way ‘traffic’ tends to force the other way out.

Synthetic perception of architectural space is potentially a powerful concept. Besides the speed and the accuracy, it could help us explicitly define essential design parameters, and critically assess our own methods of work. Although computational models are far from displaying similar performance to human perception, we can learn a lot about ourselves by
simulating some aspects of spatial behaviour. Computational experiments could reveal some fundamental rules of interaction between inhabitants and their environment, which are likely to hold true also in more complex context.

4. Conclusion

In spatial simulations the system/environment distinction helps one to leave aside ambiguities that encompass space definitions. Anything external to a system itself can be considered as its environment. The environment is made of other dynamic or static systems. Systems that are situated in a simulated environment can be simple or intricate depending what they are designed for. Although a simpler system is naturally more predictable, surprising results can be obtained from its interaction with the environment. When one models a system-environment interaction, one has to solve the problem of perception that is concerned with mechanisms how the system perceives its environment. As the environment consists of other systems, the perception of environment is defined by the way the system is coupled with other systems.

Communication within an agent population is partly also a cognitive issue. Any kind of communication needs some kind of medium that is used to interchange messages between individuals. In nature, many social insects and animals communicate through altering their environment. Since agents perceive other agents as their environment, it makes little difference whether the communication happens directly or is mediated by another the environment.

Computational simulations with agents offer an alternative approach to city planning and architectural design. In contrast to traditional top-down planning methods, agent based models are distributed bottom-up models that can be employed to analyse environments and process spatial information on locally. Observing an agent simulation could reveal some properties of the environment that otherwise might remain unnoticed. Agents that are situated in a simulated environment can interact with it from their own perspective without seeing bigger picture and could only react to immediate spatial stimuli. Agent based models can simulate certain aspects of human conduct or mimic some behavioural patterns of other urban actors. It might not be reasonable to argue that agent models can
tackle the full range of intricate relationships and complexities in the urban context; however, these models are good enough to simulate and perhaps even explain some functional patterns in the city.

Simulations with agents tend to be analytical models. However, the simulation can also be turned into a generative process where agents’ perception of the environment is used in a creative manner. The opportunity for a designer using agent based models lies in the intervention. If an acceptable abstraction of system-environment interaction has been set up, the designer can run the simulation testing out different spatial configurations, and observe the consequences of deliberately made changes in the environment.

References


Renee Puusepp
e-mail: puusepp1@hotmail.com

Paul Coates
e-mail: P.S.Coates@uel.ac.uk

CECA, AVA, University of East London
4-6 University Way, London, E16 2RD
Interactive Simulation in Virtual Environments -
A Design Tool for Planners and Architects

Joachim Kieferle, Uwe Wössner and Martin Becker