NORDISK ARKITEKTURFORSKNING
NORDIC JOURNAL OF ARCHITECTURAL RESEARCH

1

ISSUE 1 2013

NORDISK ARKITEKTURFORSKNING
Nordic Journal of Architectural Research

1–2013

Nordic Journal of Architectural Research
ISSN: 1893–5281



*Submitted manuscripts*
Manuscripts are to be sent to Madeleine Granvik (Madeleine.Granvik@slu.se), Claus Bech-Danielsen (cbd@sbi.aau.dk) and Anni Vartola (anni.vartola@aalto.fi) as a text file in Word, using Times New Roman font. Submitted papers should not exceed 8 000 words exclusive abstract, references and figures. The recommended length of contributions is 5 000–8 000 words. Deviations from this must be agreed with the editors in chief. See Author's Guideline for further information.

*Subscription*
Students/graduate students
Prize: 250 SEK, 205 DKK, 225 NOK, 27.5 Euro
Individuals (teachers, researchers, employees, professionals)
Prize: 350 SEK, 290 DKK, 320 NOK, 38.5 Euro
Institutions (libraries, companies, universities)
Prize: 3 500 SEK, 2900, DKK, 3200 NOK, 385 Euro

Students and individual subscribers must inform about their e-mail address in order to get access to the journal. After payment, send the e-mail address to Trond Haug, trond.haug@sintef.no

Institutional subscribers must inform about their IP-address/IP-range in order to get access to the journal. After payment, send the IP-address/IP-range to Trond Haug, trond.haug@sintef.no

*Payment*
Sweden, pay to: postgirokonto 419 03 25-3
Denmark, pay to: Danske Bank 1-678-0995
Finland, pay to: Sampo Bank 800013-70633795
Norway, pay to: Den Norske Bank 7877.08.13769

Outside the Nordic countries pay in SEK to SWIFT-address:
PGS ISESS Account no: 4190325-3, Postgirot Bank Sweden, SE 105 06 Stockholm

Published by SINTEF Academic Press
P O Box 124 Blindern, NO-0314 Oslo, Norway

# CONTENTS



Picture on the front cover from ByBo's housing project «Løvåshagen». Photo: Knut Egil Wang

NORDISK ARKITEKTURFORSKNING
NORDIC JOURNAL OF ARCHITECTURAL RESEARCH

# DESIGN POSSIBILITIES OF EMERGENT ALGORITHMS FOR ADAPTIVE LIGHTING SYSTEM

TONI ÖSTERLUND

## Abstract

*Algorithm aided design* methods in architectural design provide mechanisms for the use of *complex adaptive systems* that are self-regulating, bottom-up processes where global behaviour emerges from the interaction of autonomous agents. Including other systems that display emergent properties, these methods can be categorised as *emergent algorithms* and they are capable of generating organic patterns, representing the complexity and intricacy of natural systems and displaying high levels of distributed intelligence.

This paper explores the use of different emergent algorithms, such as *Cellular Automata, Lindenmayer system* and *Swarming Algorithms* in relations to an adaptive lighting system that reacts to users and changes in environmental conditions. The design of this kind of computational model requires new design methods and tools for control in enabling design exploration. A graph-based approach is presented as a possible solution for the difference between the continuous virtual coordinate space and the actual scarce placing of urban light sources. The light sources and sensors are mapped as a network of interconnected nodes and it is used in defining the topology of the virtual coordinate system. The challenges defining the functionality and enabling design using the emergent methods are explored and possible solutions are discussed.

Key words:
Adaptive Urban Lighting, Emergent Algorithms, Complex Adaptive Systems, Cellular Automata, Swarm Algorithms, L-Systems, Design tool

## 1. Introduction

*Algorithm aided design* methods in architectural design provide mechanisms for developing generative design strategies based on complex adaptive systems (CAS). CAS are self-regulating, bottom-up processes where complex behaviour arises from local interactions of multitudes of smaller autonomous components. Including other systems that display *emergent* properties, these methods can be categorised as *emergent algorithms* and they are capable of generating dynamic organic patterns, representing the complexity and intricacy of natural systems and displaying high levels of distributed intelligence. The emergent algorithms discussed here are *Cellular Automata (CA), Lindenmayer System (L-system)* and *Swarming Algorithms* (*SA*). SA are considered as a subset of a larger definition called *Agent-Based Modelling (ABS)* or *Multi-Agent Systems (MAS)*.

This paper explores the use of *emergent algorithms approach* to the modelling and realisation of an *adaptive urban lighting system*. Here, the emergent algorithms approach means the global behaviour of CA, L-system and SA, and the emerging, constantly changing organic patterns they produce; and an adaptive urban lighting system signifies the interactions with users and adaptation to changes in environmental conditions through sensors and other environmentally sensitive data. The information between the physical and virtual worlds is conveyed using sensors and lighting sources. The occurrences in the physical world are translated to the virtual through sensors and the light sources reveal the changes in the virtual configurations.

The computational model discussed here is the core of an overall system for the creative design and simulation of adaptive urban lighting. The developed system allows creativity in a way that it is not only focused on optimisation of different parameters such as lighting levels or energy consumption, but entails aesthetic and artistic design aspects as well. Adaptive urban lighting is approached from the viewpoint of experience of the pedestrian in artificially lit urban environments (Pihlajaniemi et al., 2012). For design ability purposes, a graph-based approach is introduced to control the design and to situate the system by mapping the lights and sensors as nodes. In connection with a graphical design tool, the control ability of the method is important as it allows for the easy design of various adaptive lighting scenarios and not just one optimised solution. Designers using the tool should be able to manage and control the design parameters in a way that they can achieve desired aesthetic and functional results.

Research questions arise from the need to design and simulate the functionalities of a complex and adaptive system and are two-fold. Firstly, the challenge is to identify and categorise different algorithmic design methods and their control mechanisms that could be utilised in the creation of an adaptive lighting model. As this describes the need from the computational point of view, which relates to the used process and its intricacies, the use of those methods as a part of the design process has to be taken into consideration. This leads to ponder about the nature and functionalities of the design tool that could aid in the design exploration (and simulation) of adaptive urban lighting.

The paper is based on an on-going research project that aims to address designers' needs by identifying suitable computational methods, categorising their control mechanisms and developing tools for their design. The emergent algorithms approach allows for rule-based, yet unpredictable and naturally dynamic behaviour of light. The emergent algorithms discussed here have shown vast potential in designing and controlling creative systems based on natural processes. They have been used in creative design projects, art and dynamic lighting installations.

### 1.1 Ambient Intelligence

*Ambient Intelligence* system describes environments that are endowed with a large number of interconnected sensors, able to perceive and react to the presence of people. It can be viewed as autonomous entities interacting with surroundings in order to obtain the desired overall system behaviour (Shadbolt, 2003; Bandini et al., 2009a). This involves bringing together several computing areas: *ubiquitous computing* (Weiser, 1991), intelligent systems research and *pervasive computing* (McCullough, 2004); the awareness of context and appreciation of social interactions. Pervasive computing describes the challenges of ubiquitous computing, not only from the technological but from cultural points of view as well. With the use of ubiquitous technology integrated into our environment, digital networks are no longer separate from architecture; digital systems embedded into our environments allow people to interact remotely, asynchronously and indirectly (McCullough, 2004, pp. xii–xvi).

Thomsen (2009) discusses the essential challenges in designing adaptive architecture and environments. He describes them as *performative* where performativity is understood as the actions between an adaptive architecture and the surrounding environment through feedback loops. This opens the possibility for each actor to engage with the designed landscape through bottom-up processes: a time-based process with interactive contacts between all actors and a possibility to include complex parameters and influences in the design process (Thomsen, 2009, p. 37). Adaptive urban lighting can therefore be considered as an ambient intelligent environment: the lighting interacts with people through adaptive feedback loops using computational bottom-up processes. The adaptation process can occur according to the presence of people using sensors or through their explicit interaction with the system using different electronic devices and interfaces. The interaction can happen through locally perceived changes in levels of illuminance, colour, distribution of light or in the rhythm of these changes. The benefits of adaptive urban lighting are connected e.g. with energy savings, traffic safety, the feeling of security of pedestrians, and with the experiential value of an artificially lit environment (Pihlajaniemi, Österlund and Herneoja, 2011).

## 2. Algorithm aided design methods

*Algorithm aided design* is a general name for a design method where computation is utilised in creative systems through algorithmic logic and generative processes. The use of computation relies on the designer's awareness and skills in computer programming and mathematics as the design process is expressed through algorithms written in computer code. This creates a large contrast against the visual way in which architects are accustomed to design, although programming is given more attention in architectural education (Fricker, Wartmann and Hovestadt, 2008; Wurzer, Alaçam and Lorenz, 2011). In the field of architectural design, the design methods implementing computation are also referred to as *algorithmic architecture* (Terzidis, 2006) or *computational design* (Menges and Ahlquist, 2011; Peters and De Kestelier, 2013).

An *algorithm* is a finite list of exactly defined instructions for completing a task and as such can represent any rule-based activity. In that sense, algorithms can be viewed as an externalisation of the design thought process, as design choices and decisions are externalised into the readable and computational form of an algorithm. However, it is notable that computation differs from the act of *computerisation* where designs already conceptualised in the designer's mind are digitised and stored on a computer system for further manipulation (Terzidis, 2006; Coates, 2010, pp. 1–4). Menges and Ahlquist (2011, p. 13) define *computation*, in relation to design, as the «*processing of information and interactions between elements, which constitute a specific environment, the pivotal word being interactions*» as Terzidis (2006, p. xi) states it as *«the exploration of indeterminate, vague, unclear, and often ill-defined processes»*.

Algorithm aided design methods inevitably alter the design process towards dynamic systems, where the designer is more guiding the process than being in total control of the outcome. This adds to the unpredictability of the design process, as the computer is introduced as a generative aid and a collaborator through computation. The generative process that defines the outcome becomes more important than the single design outcome itself (Frazer, 1995, pp.102–103; Terzidis, 2006).

### 2.1 Theoretical framework

Based on the concepts and principles of *general system theory* (Bertalanffy, 1969), c*omplexity science* as a theoretical background tries to understand and explain these kinds of systems where large networks of components operate without central control. Complexity science deals with phenomena such as self-organisation, evolution and adaptation and tries to explain with simple rules the emergent, unpredictable bottom-up characteristics of these *complex systems*. It concerns with the «wholeness» of the system that is not understandable by the isolated inspection of their parts, but through the global behaviour it exhibits, emerging from the collective interactions of many components (Waldrop, 1992; Kaufmann, 1995). This is related to the *dynamical systems theory (dynamics)*, which studies the description and prediction of complex systems over time, trying to find unchanging laws that generate changing configurations (Kauffman, 1993, pp. 175–181; Holland, 1998, pp. 45–46; Mitchell, 2009, pp. 15–16).

*Complex adaptive systems (CAS)* consist of large networks of individual components that operate under very simple rules with no centralised control. The collective actions of the components emerge as complex, hard-to-predict behaviour, sophisticated information processing and adaptation via learning or evolution (Holland, 1992; Waldrop, 1992; Mitchell, 2009, pp. 12–13). *Emergence* refers to the macroscopic behaviour and the production of forms of a complex system; a system that cannot be deduced from its components. It is more than the sum of its parts, as the 'whole' emerges from the multitudes of successive interactions of its elements (Holland, 1998; Weinstock 2010, pp. 31–32).

Model of a system that is based on autonomous components that are governed by their individual rules is called an *agent-based model (ABM)*, where the individual 'components' are known as *agents*. As the interactions between agents increase, the system demonstrates sophistication well beyond the very simple rules of the individuals (Holland 1998, pp. 116–124). When the model consists of multiple acting agents that interact with each other and also react to their environment, it is known also as a *multi-agent system (MAS)* (Jennings, Sycara and Woolridge, 1998). There is not a clear distinction in science between using the term ABS or MAS (Niazi and Hussain, 2011), but in general terms, the former is used in modelling human social and organisational behaviour, as the latter is used in research concerning the design and development of 'artificial' agents (Macal and North, 2009).

## 2.2 Tools for designing complexity

Designing complex adaptive and interactive lighting control systems using emergent algorithms is challenging as the system demonstrates global behaviour that is very hard to predict and control. The challenge of top-down design of the local interactions for a desired global behaviour is a significant research question in ABM/MAS (Yamins and Nagpal, 2008; Bandini et al., 2009b; Essen, Offermans and Eggen, 2012) and in the theory of L-systems it is called *inference problem* (Prusinkiewicz and Lindenmayer, 1990, p. 11).

In real-world settings where a light reacts to or even interacts with human actors (Thomsen and Poulsen, 2010; Pihlajaniemi et al., 2012), the local behaviour and its mechanics become an essential part in the design of the system. Therefore the creative process and design tools should allow for the easy exploration of different local behaviours through parameter configurations in an intuitive way (Poulsen, Andersen and Jensen, 2012; Pihlajaniemi, Österlund and Tanska, 2012). This requires real-time simulation of the local and global level behaviours that the manipulations of design parameters induce. The exploration of design possibilities should happen visually and the design tool should facilitate this need. Vrachliotis (2007) argues for the game-like qualities of such systems since the designer is placed in the midst of a real-time visualised simulation and given the power to guide the process through interaction, yet not being in total control of the emergent nature of the 'game' (Borries, Walz and Böttger, 2007, pp. 322–343).

Textual and visual scripting is becoming increasingly integrated into our current design packages and environments (Aish, 2013). Current CAD and 3D modelling software such as *Rhinoceros* (McNeel, 2013), *Maya* (Autodesk, 2013a), *3ds Max* (Autodesk, 2013b), *Blender* (Blender Foundation, 2013) and AutoCAD (through *DesignScript*) (Autodesk, 2013c) can be used to visualise and control the behaviour of dynamic systems through scripting. However, when utilizing these software, the system functionality and control methods are bound to the software's environment and user interface and cannot readily be extended as stand-alone or specialised applications (Bandini et al., 2009a).

NetLogo (Wilensky, 1999), StarLogo (Resnick, 1994) and Repast Simphony (North et al., 2013) are examples of tools that are used specially in modelling and simulating agent-based systems. They are designed to be simple to be adopted and they are used to teach agent modelling and prototyping basic agent behaviour as well as in performing limited analysis (Macal and North, 2009).

General purpose programming toolkits or IDE's (Integrated Developing Environments) for visual and interactive content, such as *Processing* (Reas and Fry, 2007), *OpenFrameworks* (2013), *vvvv* (2013) or *Cinder* (2013), offer more flexibility in creating dedicated graphical design tools and user interfaces. However, they do not readily offer specialised tools for modelling agent behaviour, but are easy to learn and extendable through freely distributed libraries. They are based on existing programming languages, such as Java and C++ and they can easily be extended from computational experiments to stand-alone applications.

## 3. Exploring design possibilities

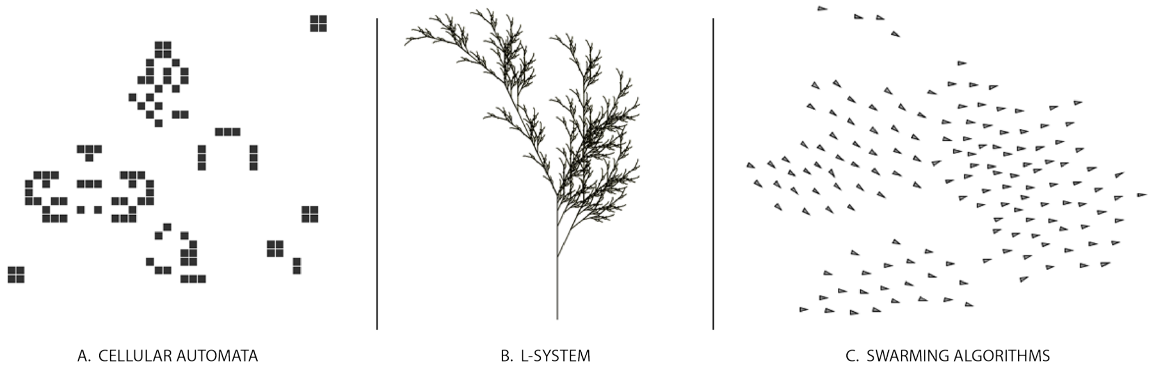



Figure 1 A–C. The different emergent algorithms in their conventional representation: A. Cellular Automata implementing the Game of Life rules, B. a two-dimensional fractal tree modelled using a bracketed L-system and C. the Boids animation with swarming algorithm.

A cellular automaton (plural: cellular automata, CA) is a modelling system, which consists of a regular grid lattice of cells in one, two or three dimensions. At any given time, each cell assumes one of a finite number of states based on transition rules, which determine the state of each cell in relations to its neighbours. Time advances in discrete steps and on every step the transition rules are applied to the whole grid simultaneously. In a simplest form of CA, the cells have only two states (1 or 0, true or false) and even with simple rules, it can produce complex emergent behaviour. The best know CA is the Game of Life, developed by John Conway in 1970, which is a two-state, two-dimensional CA that simulated seemingly life-like behaviour (figure 1 A). Cellular automata were originally devised by von Neumann and Ulam in the 1940's to describe organic self-replicating systems (Frazer, 1995, pp. 51–55; Wolfram, 2002).

The *Lindenmayer system* (L-system) is a grammatical rewriting system, first introduced by Lindenmayer in 1968 as a method for modelling growth patterns of algae and bacteria. The L-system was later extended to include branching topology on higher plants (also known as *bracketed L-systems*) (figure 1 B). The L-system consists of two parts: a generative and an interpretive process. The generative rewriting process produces a string of letters that has to be interpreted as actions for an effective representation. Graphically this is done by implementing *turtle graphics* (Resnick, 1994): a concept of a path drawing turtle that moves and turns in discrete steps according to the symbols. Each symbol represents a different action in the turtle's movement. L-systems follow recursive rules and resulting forms exhibit self-similarity and fractal geometry (Prusinkiewicz and Lindenmayer, 1990).

Swarming algorithms (SA) are the collective behaviours of decentralised particles acting out simple local interaction rules in two or three-dimensional space. They were developed by Reynolds (1987) for his Boids animation (figure 1 C). The swarm members navigate according to simple rules in reaction to their environment and each other, which emerge as complex global behaviour that mimics the flocking behaviour of birds. Swarm algorithms are essentially MAS where individual intelligent agents interact between themselves and the environment. It is a decentralised and self-organising system with no central controlling agent, but the global behaviour emerges from the local interactions of the swarm members.

### 3.1 The similar properties of the emergent algorithms

Viewed in detail, the canonical representations of CA, L-system and SA are quite different, and by looking at the geometric patterns and compositions they form, easily distinguishable. The main visible difference is in the system's mobility: the CA cells are bound to their location and interact by reacting to their neighbours' states, whereas SA particles move freely and interact by reacting to their neighbours' locations. However, when viewed in their abstract level, they interestingly start to display common characteristics. The cells in CA and the swarming particles in SA are both essentially *agents* and the common nominator is the interaction between an agent and its local neighbourhood. In both, local interactions affect the global configuration of the system.

When the visual representation methods generally associated with CA and SA are peeled of, even a small addition or alteration to the basic approaches can render them undistinguishable from another, which leads to hybrid solutions (Bandini, Federici and Vizzari, 2007). The use of these complex methods and the exact manner in which they are implemented depends on the properties and needs of the model they are used to simulate (Partanen and Joutsiniemi, 2007).

Although L-system as such is not considered to be a complex system, the interpretation process of the generative string of symbols can also be viewed as a computational rule of a decision making agent (Prusinkiewicz, 2000). On the other hand, Wolfram (2002, pp. 82–87) studies L-system as a recursive substitution system, where a single character is substituted with a string of characters. He experiments in different graphical substitution systems, where a cell with a certain state is substituted with a larger set of cells (in various states) according to the substitution rules. Wolfram is able to simulate CA type of behaviour using these substitution systems and as such brings the concept of L-system closer to the concept of the CAS (Prusinkiewicz and Lindenmayer, 1990). As a rewrite system with substitution rules, the L-system is also similar to *shape grammars* (Stiny, 1980; Mitchell, 1990).

### 3.2 Mapping the virtual and real spaces

The emergent algorithms work within a continuous virtual space and reveal the states of the individual components through changes in location, geometry or colour. The virtual coordinate system is often not limited and the different states of the components can be visualised in every possible location (figure 2). If the output is displayed on a computer monitor, it has significant difference in resolution compared to the limited amount and scarce placing of the physical light sources in a real-world setting. This reveals the founding difference in the way that the conventional versions of the emergent algorithms operate in virtual coordinates and the actual placing of urban light sources. The disposition and spacing of the light sources may vary drastically and the sources themselves may differ in use, orientation, height and illumination.

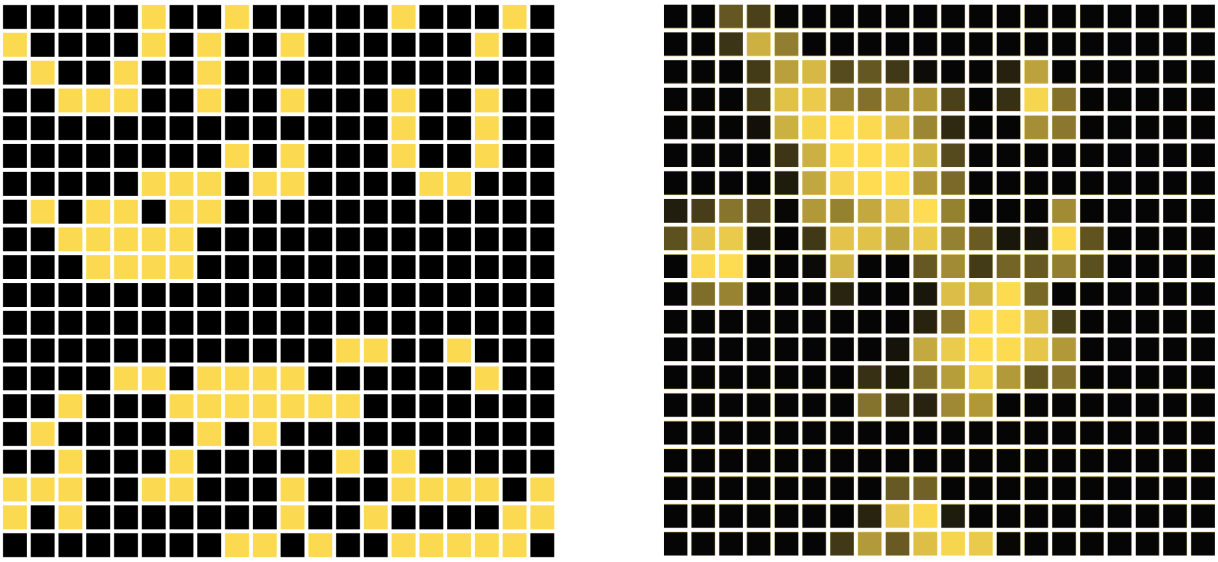


Figure 2. A regular 20 x 20 pixel grid coordinate space. On the left, Cellular Automata patterns implementing Game of Life rules. On the right, a swarming algorithm in operation, where the distance to the agents affect the colour intensity of an individual cell.

The locations of the light sources can be superimposed on the virtual coordinates. However, as they are the only physical outputs of the changes in system states, the challenge is to control the emergent process in a way that the lights get 'activated'. The behaviour of the process is unpredictable and it is impossible to guarantee that the virtual states corresponding to the physical lighting locations show desired interactions. This could lead to an extreme situation where the process is not visualised at all by the light outputs.

The resolution of the virtual space affects the process. Defining lower resolution can decrease the complexity of the process as it confines it to a smaller space or limits the amount of interacting components (figure 3). Affecting factors that have to be taken into consideration can also include the shape of a cell, the global bounds of the grid and the heights of light sources. Bandini et al. (2009b) describe a possibility to define the resolution of their dynamic visualisation tool by a subdivision approach, where the area of a single cell (a light group) can be further subdivided in 9 sub-areas (9 different light groups).

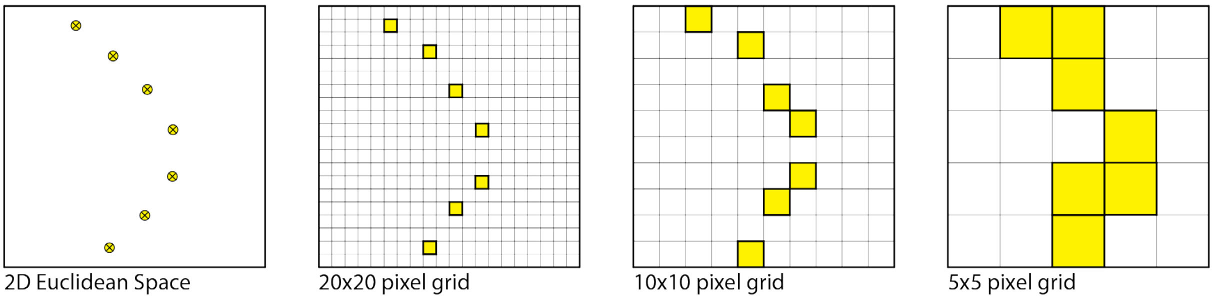


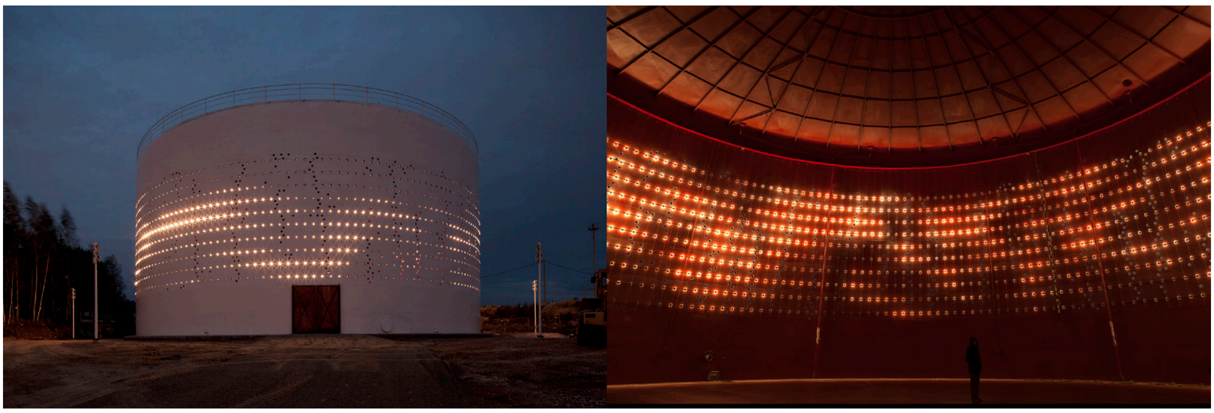

Figure 3. If the lighting sources are superimposed on a virtual coordinate system (here, the regular grid lattice of a cellular automaton), the size and orientation of the grid has an effect on the accuracy and resolution of the computational process.

In examples of light art installations that are implementing these algorithms, the popular solution has been to populate the real-world space with a sufficient amount of light sources in a way that it represents the coordinate system and resolution of the virtual space. This requires an abundance of lights to accurately depict the pattern formation of the emergent methods, where a single light source represents a single point in space or a cell in a grid.

The London based design studio rAndom International (2010) created a responsive light installation called *Swarm Light* consisting of 3000 LED modules in three 10x10x10 cubes of 81 centimetres and four microphones that pick up sounds from the audience. Also the interactive light installation called *Flylight* by Studio Drift (2011) uses a swarming algorithm, but the minimum of 180 glass light tubes are not in a grid formation; their positions and directions themselves mimic the pattern of a swarm. The virtual movement of the swarm happen within the borders of the installation, but otherwise in an open and unobstructed virtual space. Lighting Design Collective (2011) won a competition to realise a light art installation in Kruunuvuori, Helsinki, Finland called Silo468. Their plan transfers an old oil silo into a light display, where 1250 LED's are controlled by a swarming algorithm (figure 4).

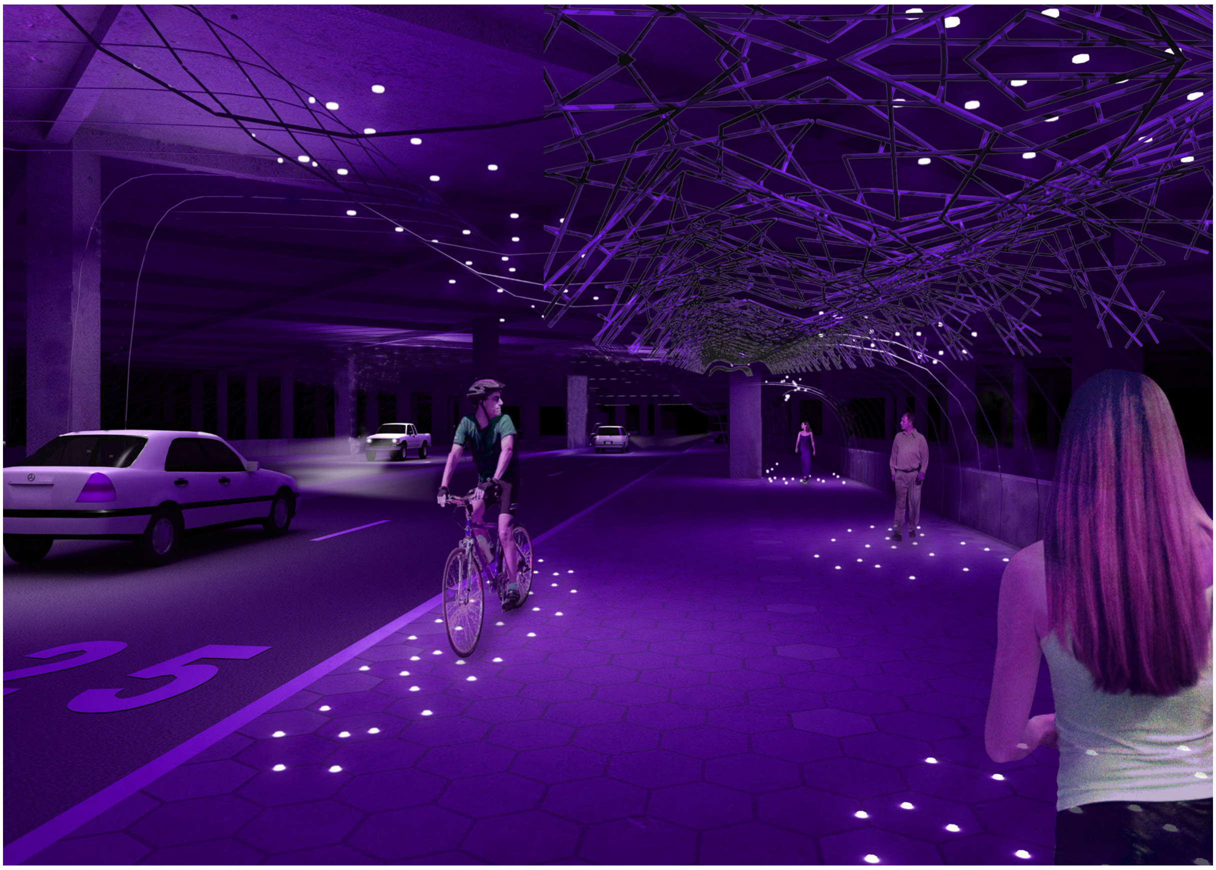
Figure 4. Silo468 by Lighting Design Collective.

PHOTOS BY TAPIO ROSENIUS.



Villareal (2009), an American artist working with LED lights and programming in order to create illuminated displays, used in his *Multiverse* sculpture for the National Gallery of Art in Washington, approximately 41,000 LEDs to occupy a space the length of 61 meters (200 ft). Although not implementing complex computational systems, Seitinger (2010), in her Ph.D. thesis, *Liberated Pixels*, blurred the idea of illuminated displays, by creating actual movable «pixels» of light that can be placed flexibly in any configuration on any surface in the city. People could interact with the units directly by moving them, through sensors or by SMS messages Bandini et al. (2009a, 2009b). researched the use of a CA based lighting installation for a New York based architectural office Acconci Studio (2011). The project resulted in a public installation in Indianapolis: *Swarm Street* uses 1000 LEDs embedded onto the pavement and another 1000 installed in an open steel frame above the street (figure 5).

Figure 5. Swarm Street by Acconci Studio (V.A., Nathan DeGraaf, Jono Podborsek, Eduardo Marques, Ezio Blasetti).



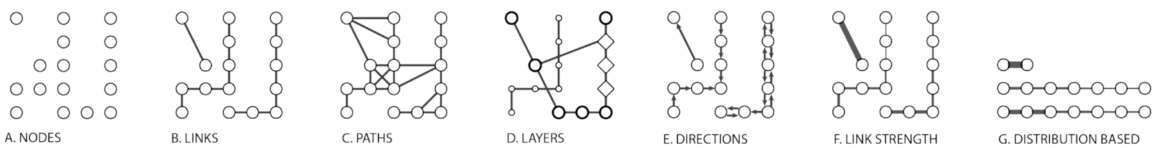
## 3.3 Graph-based coordinates

There are various methods within the different complex system morphologies that enable the handling of irregular distribution of output cells. One of their shared properties is the notion of using networks instead of regular coordinate grids. In CA, this is referred to as *graph-CA* (O'Sullivan, 2000) or *Automata Network* (Bandini et al., 2009b). In agent-based systems, the network based approach can be described as *Multi-Agent Situated System* (Bandini, Manzoni and Simone, 2002). The term *situated* refers to the mapping of the nodes in correspondence to the locations in physical space. If the individual nodes are differentiated, which means that there exists different types of nodes where each type has their own set of rules, the methods can be called multilayered, such as *Multilayered Automata Networks* and *Multilayered Multi-Agent Situated System*. In multilayered systems, the graph can also have a hierarchical structure based on nested networks, where nodes in a network can themselves be a nested network of lower level (Bandini et al., 2009b).

On L-system using turtle graphic interpretation (Resnick, 1994), there are two methods that use graph terminology: *edge rewriting* and *node rewriting*. These are similar to the string rewriting but they rewrite the geometric representation itself. The *Map L-system* uses planar graphs with cycles, called maps, to express the topology of the resulting string. (Prusinkiewicz and Lindenmayer, 1990, pp. 11–18, pp. 145–153).

The proposed solution for the coordinate problem is to consider the lights as a network visualised as a graph (figure 6). The single node on the graph represents either a light source or a sensor (a), which can be linked together to form networks and to define clustering of nodes (b). The connections define the topology of the network and the communication pathways, by which the component interactions happen (c). The different types of nodes can be visually differentiated using symbols and as such, multiple computational layers can be formed (d) with each layer having its own distinct interaction rules. The links can have direction, adding the possibility for defining unidirectional or bidirectional connections (e). The link strengths can also have a meaning within this system and can be visualised, for instance, with line thickness (f). The graph can be mapped on top of a visual representation of the physical space to better reveal their locations. Alternatively, it can be distributed based on the network topology, which gives a clearer presentation of the relationships between different nodes (g).

Figure 6. Graph representation of lighting networks.



### 3.4 Situated emergent algorithms

The CA as a regular grid can be easily visualised as a graph. This method in visualising local connections of a single cell is quite commonly used to distinguish the number of neighbours each cell takes into account when computing its next state (figure 7). In the *von Neumann neighbourhood,* each cell considers only the four adjacent neighbours, whereas in the *Moore neighbourhood* the diagonal neighbours are also taken into account. The links do not have direction or weights, only their number and the states of the connecting nodes are considered.

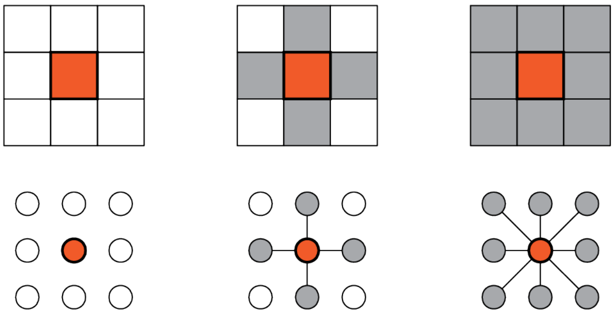


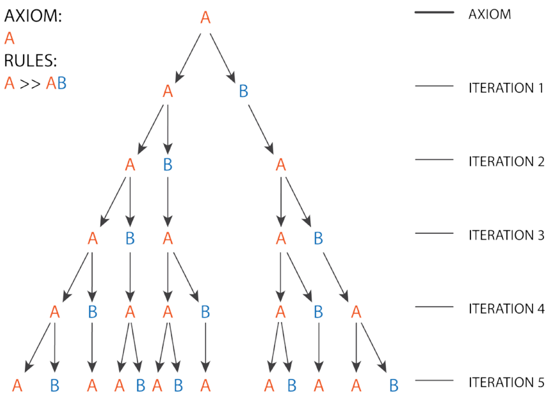
Figure 7. Cellular Automata grid and neighbours presented as graphs. From the left; the active cell under inspection, von Neumann neighbourhood and Moore neighbourhood.

Bandini, Federici and Vizzari (2007) provide an example of using situated CA in a research to model crowd dynamics. They call their method *Situated Cellular Agents*. The cellular lattice and its topology is a rough representation of the simulated space and the occupancy (or state) of the cell is visualised by the cell's colour. The used model provides two interaction mechanics; neighbour interaction and a field of emission. This kind of approach modifies the basic CA approach: it is representing more of an agent-based model where the agents move in a discrete cellular space. The cellular space is represented as nodes with undirected links. It is notable that the amount of connections vary according to the node location. The network is fixed and scale-dependent; the size of the single cell is 40 x 40 cm.

Partanen and Joutsiniemi (2007) used *graph-CA* (O'Sullivan, 2000) in simulating the complex behaviour of building site usage change over time. The graph approach was selected, because it represents the irregular structure of the sites better than a regular grid lattice. The interaction rules for the CA were controlled by a matrix of factors describing the compatibility and desirability between different site usages. The use of the factor matrix increased the controllability of the simulation because it allowed the review of individual neighbour relationships instead of defining uniform interaction rules.

Bandini et al. (2009b) describe the use of *Dissipative Cellular Automata*, which is an extension of the conventional version of CA. Cellular automata is closed and synchronous system that has no mechanism for sensing external stimuli (closed) and each cell's state is updated at the same time on discrete time-steps (synchronous). In Dissipative Cellular Automata, the cells update their states asynchronously and can be directly influenced by the external environment. This notion was further refined to include the multi-layered approach of CA in system called *Dissipative Multilayered Automata Network* (Bandini et al., 2010).

It is evident that the notion of conventional CA can easily be extended to include multiple states and layers and be calculated over an irregular network of nodes. The distinction between different complex systems is very difficult since it is easy to extend the functionality of the system to match the needs of the simulation and create hybrid solutions (Bandini, Federici and Vizzari, 2007; Partanen and Joutsiniemi, 2007). The individual cell in CA easily converts into an agent if the interaction rules are extended to e.g. include movement or communication beyond their neighbouring cell's states.



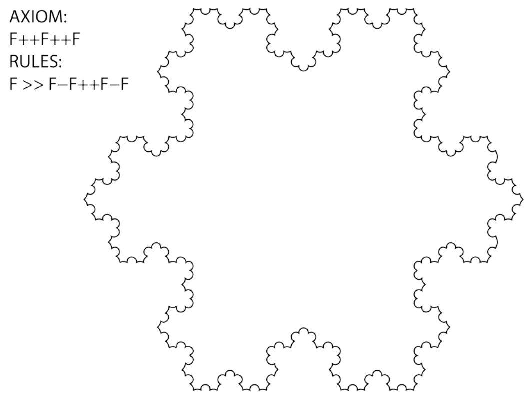




Figure 8. The rewriting process of the L-system is operated through iterations where the characters are substituted with other characters according to the rewriting rules. For instance, the Koch snowflake can be drawn using the L-system and turtle graphics interpretation. The resulting image, on the right, is the fourth iteration with a turn angle of 60 degrees.

The basic core of the L-system is a recursive substitute system and it can be used with many different graphical representations (Wolfram 2002, pp. 82–87). As the L-system functions on the level of substituting letters, it can be noted that the L-system as such does not deal with coordinates or space (figure 8). Another form of graphical interpretation process is the *Map L-system*, where the string is defined using a class of planar graphs with cycles, called maps. This creates cell topologies, where the neighbourhood relationships are established, but the geometry remains unspecified. This method was originally developed to model cellular development in morphogenesis (Prusinkiewicz and Lindenmayer, 1990, pp. 145–153). Hansmeyer (2003) has conducted extended geometrical experiments with the Map L-system, where he has mapped symbols or characters to geometrical object properties (such as scale, rotation or location) and mesh surface vertices (which are moved according to the rules). The

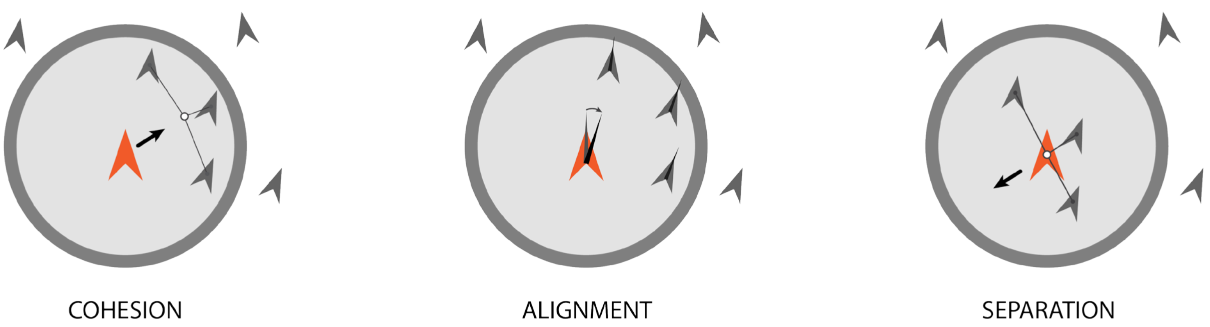
Genr8 surface design tool, developed by Hemberg (2009) as part of the Emergent Design Group at MIT, uses evolutionary search and a version of the Map L-system. The Genr8 uses evolutionary search to discover L-system rules that adaptively evolve towards a surface with features the user has specified.

The use of L-system in graph-based representation of adaptive lighting becomes possible by interpreting and mapping the resulting string to the nodes as parameters for the light sources. The challenges arise from the design of the interpretation rules and actions that the different symbols produce. On the other hand, it allows for a wide freedom in the design of the system's functionality.

As the adaptive system is constantly changing through external stimuli via the sensors, the L-system should also have reduction rules (Coates 2010, p. 73) that instead of lengthening the resulting string gradually shorten it by replacing a defined set of characters with a shorter defined set. The iterations of the string substitution process can be visualised either by discrete time-steps where the string is rewritten synchronously (as ticks of a virtual clock), or asynchronously as a reaction to local environmental factors. Several L-systems can simultaneously be calculated within the same graph, thus producing overlapping effects.

In conventional swarm algorithms (Reynolds, 1987), the graph-based representation of the virtual space is not possible as the swarm moves freely and unconstrained following simple steering rules (figure 9). Conventional SA can be expanded by additional steering behaviour, such as the path following algorithm, explained by Reynolds (1999), where the agents steer to keep close to the path. Using the graph-based coordinate system, the swarming behaviour can also be demonstrated by constraining the agents' movements to happen strictly within the node topology, which limits the movement to basically one-dimensional space (from node to node). The virtual agents can still move freely within the constrained space of the graph and interact with each other and the environment.

Figure 9. The local rules of swarming agents, according to the Reynolds's Boids animation (1987).



Using an agent-based system in creating an autonomous lighting control, Essen, Offermans and Eggen (2012) state that in this context agents can represent light sources, different sensors and people. In their approach however, by mapping graph nodes to the physical locations, the nodes becomes situated. This effectively fixes the locations of the agents, allowing them to communicate with each other and the environment, but not to move or relocate themselves according to the swarming rules. This method has similarities with the Situated Cellular Agents system (Bandini, Fedrici and Vizzari, 2007).

In an effort to maintain the fluent nature and emergent behaviour as the conventional SA, the situated light source nodes cannot be considered as agents, but the nodes should only act as output devices reacting to the agents' locations. Individual agents have properties, such as signal strength and an emission fields that activate the light sources and accumulate in dense swarms. The agents themselves can move freely within the space defined by the network and information about the environment and people is received through the situated sensor nodes. The sensor data can offer stimulus to the swarm which behaves according to their local rules and creates emergent global behaviour that is visible through the changes in light sources.

Table 1. Examples on the benefits and challenges of using the different emergent methods with a graph-based coordinate system to design and control adaptive lighting.

| *Method* | *Benefits* | *Challenges* |
|---|---|---|
| **Cellular Automata** | • Can be readily represented and calculated using graph-based coordinates.<br>• The state of the cell is directly mapped to the output of the light.<br>• Transition rules can vary within and between different cell types.<br>• Possibility for asynchronous update and external stimuli. | • Limited amount of design parameters to control.<br>• Difficult to see the correlation between the manipulation of design parameters and the changes it has in the system.<br>• Updates in discrete time steps. |
| **Lindenmayer System** | • Different interpretation mechanisms can be created for the generative string.<br>• Allows for many kinds of different control parameters through the interpretation rules.<br>• Possibility for reduction rules.<br>• Possibility for asynchronous update. | • Conventional graphical interpretation cannot be used.<br>• Challenging design of the interpretation rules.<br>• Difficult to see the correlation between the manipulation of design parameters and the changes it has in the system. |
| **Swarming Algorithms** | • Creates fluent, organic and gradient dynamic patterns.<br>• Allows for many kinds of different control parameters in steering, speed, strength, etc.<br>• A good correlation between the manipulation of design parameters and the changes in the system. | • No direct mapping between the light nodes and agents.<br>• Constraining the movement to the graph-based coordinate system. |

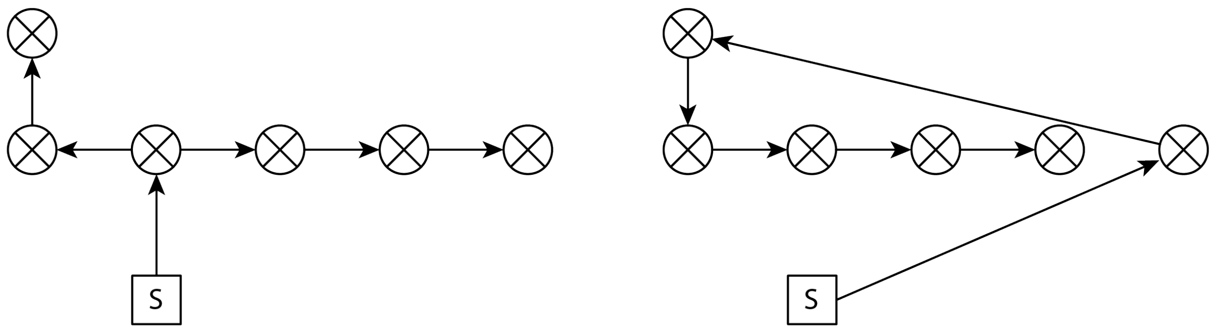
## 4. Interface possibilities in aid of design exploration

Design exploration with emergent algorithms for a desired local and global behaviour is conducted through the manipulation of several interconnected control parameters. The different parameter configurations affect the local interaction rules and thus the global behaviour of the system. Even though the Processing platform does not require compiling, the parameter manipulation through code is cumbersome and unintuitive. Thus the design exploration needs to be implemented on graphical user interface (GUI) level in order to ease the design decision making.

The graph-based representation of the lighting and sensor network does not make assumptions on the detailed functionality of the emergent algorithms. The nature of the agent or other methods used will determine whether the nodes are situated agents at fixed locations or whether the complex system can navigate freely within the bounds of the network coordinate space. In other words, the nodes (lights) could act as situated autonomous agents that interact and change states (e.g. intensity, colour), but do not move, or they could function as unintelligent outputs that only visualise the state and location of the system's individual agents. Such a demarcation of the detailed mechanics of the system is not made at this point of the research.

When implementing light sources and sensors as networks, the definition of the topology becomes an important design task (figure 10). It allows extending connections beyond physically adjacent light sources, thus inducing reaction at longer distances. The absence of a connection carries an equal amount of importance since it allows the topological separation of adjacent light sources. Exploring different topological configurations could have a big influence in the way that the light patterns move within the system, even without changing any other design parameters.

Figure 10. Defining network topology becomes part of the design process.



The graph can be visualised on top of a zoomable map, showing the actual locations of the nodes (figure 11). This map visualisation corresponds with the real-life effects the adaptive system creates, but may not be the best type of visualisation to define the connections. Depending on the scale, group of nodes may be cluttered, which make it challenging to connect individual nodes.

Figure 11. Preliminary tests in graph-based representation of the adaptive urban lighting environment. The node colours indicate the number of links and the label identifies each light source with a unique id. The graph topology is not generative, but a deliberate design decision. This picture does not represent any actual design situation, but is merely a test made for demonstration purposes.

The network can also be visualised in a way that the nodes are organised based on the topology of the graph and not by the mapping to their physical locations (figure 12). This kind of layout may offer insight into the connections and the operation of the adaptive system within different layers. Nodes can be moved manually to better read the connections between nodes, but it may not be the best option when operating with a large and interconnected network. Using the force-directed layout approach where the nodes self-distribute based on intrinsic *spring forces* of the links (Heer, Bostock and Ogievetsky, 2010) may provide better results in complicated networks. It offers a computational method of untangling the nodes and also reveals clustering. The visual representation of the topology should not be limited to any single representation, but should offer fluent transition or even simultaneous views of different layouts.

The design tool should allow easy design exploration and parameter manipulation in a way that its effects can be comprehended from the global behaviour of the system. The emergent algorithms have many different controllable local interaction parameters that have profound effect on the global behaviour of the system. The addition of layers; by defining various types of nodes or various types of agents, increases the number of controllable parameters. This adds to the complexity of the design process, but also offers increased possibilities. Agents can have different rules based on their type, where they interact differently with each other than with agents of other type.

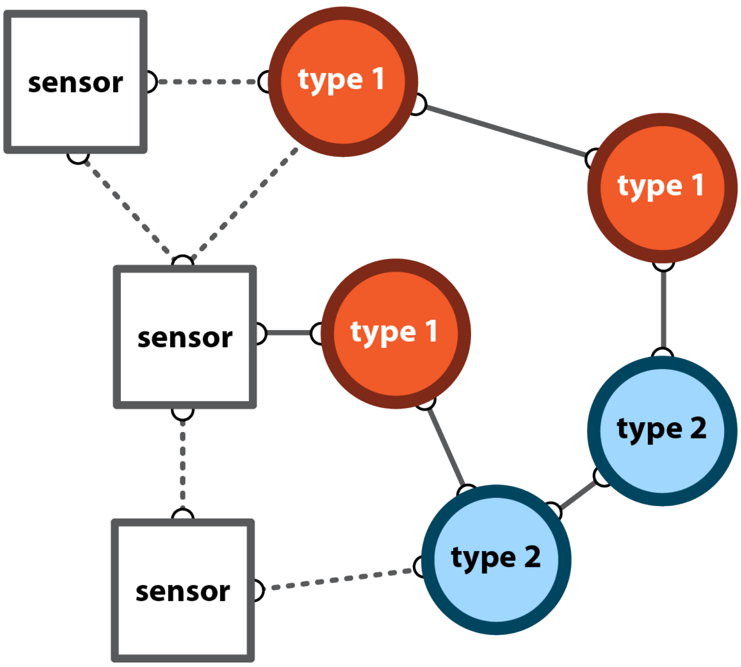



Figure 12. Network distributed in a way that does not correspond to the node's actual locations, but offers easier readability of the graph topology.

The parameter manipulation can happen directly through the graph by selecting relevant nodes or other visual elements with a cursor (figure 13). This way, only the relevant portion of the controllable parameters can be manipulated at once. This kind of additional mode for control would allow for the extensive manipulation of the different parameters on lighting and sensor nodes and their effects on the global behaviour can be perceived in real-time. This approach can also be extended for the mouse-based manipulation of node topology and link strengths.

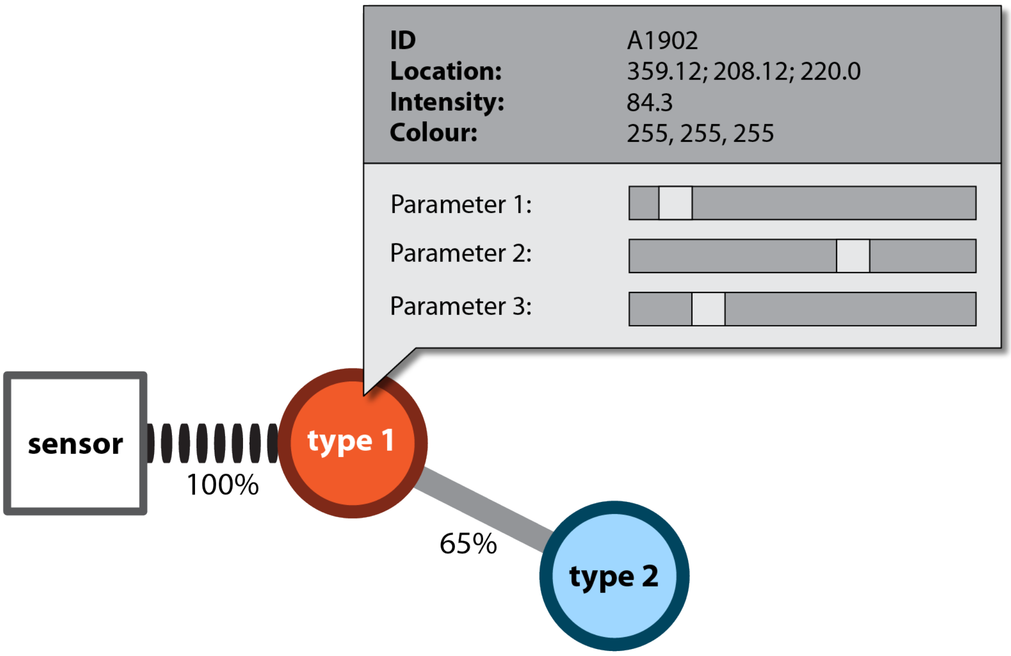



Figure 13. An individual node or other visual elements can be selected using the mouse; additional controls are displayed and it becomes possible to manipulate relevant parameter. If different node types (layers) exist, the parameters may vary according to the requirements and rules associated with the node type.

## 5. Future development and conclusions

This paper has discussed the use of different emergent algorithms, such as CA, L-system and SA in relations to an adaptive urban lighting system that reacts to user actions and changes in environmental conditions. The computational model utilises methods of distributed intelligence and the bottom-up control of the local interaction rules in order to find desirable global system configurations. The emergent algorithms approach allows for rule-based, yet unpredictable and naturally dynamic behaviour of light. Their use in design is challenging as they increase the complexity of the design process, though they offer new exciting possibilities for controlling adaptation.

The system can be harnessed in an ambient intelligence environment i.e. a physical environment endowed with a large number of interconnected sensors that react to the presence of people. The environment is in constant interaction with its users and changes in environmental conditions, through feedback loops and bottom-up processes. The design of a computational system exhibiting this kind of behaviour requires new design methods and tools. The methods discussed in this paper allow for easy design and control of adaptive processes in urban lighting. The benefits of using adaptive systems in urban lighting include energy savings and increasing the feeling of security and experiential value of artificially lit environments.

A graph-based approach is presented as a possible solution for the difference between the virtual coordinate space and the actual scarce placing of lighting sources. The graph is used to map the light sources and sensors as a network of interconnected nodes. The network topology itself can be freely defined by the designer. This expands the control possibilities because it provides a new level for design that is not possible to achieve with a regular coordinate system. The graph-based system offers extended control and design possibilities even with more simple and straightforward control algorithms that may not implement emergent methods.

Conclusively, this paper suggests that the improved design ability of emergent algorithms for adaptive lighting can be achieved without restricting the global distributed approach. The designer can define the topology of the graph and distribute the nodes freely for a better visualisation of the network. Agents can have different types with their own distinctive behavioural rules, which create multiple computational layers. All manipulation of agent's control parameters can be done via the graphical user interface using the mouse and there is no need to touch the underlying code of the system. The presented design tool interface possibilities can assist in the design and control, but they do not diminish the unpredictable nature of the emergent algorithms.

Future research should aim to develop the presented ideas further into a functional design tool prototype. For this purpose, a design tool is in the works: it simulates and visualises these adaptive processes in a stylised 2D environment, representing an abstraction of the lights' physical environment. The development of this tool is done within and for the needs of the Adaptive Urban Lighting – Algorithm Aided Lighting Design research project (AUL) at the University of Oulu, Finland. The programming of the tool is done within the Processing environment and the tool's functionalities have been designed within the AUL research team and tested in real-world design cases. The design tool allows for the manipulation of environmental factors (such as time of day and time speed, temperature, sky illuminance, surface luminance, etc.) and the testing of different control algorithms.

Future research questions concern about the selection and development of the emergent, agent-based model and the definition of the different control parameters. As the desired functionalities and methods presented in this paper give guidelines to the implementation, the actual construction of the graphical design tool and its detailed functionality is still a challenge on its own.

## 6. Acknowledgements

The research has been carried out in the University of Oulu, Faculty of Technology, Department of Architecture, Finland as a part of the *Adaptive Urban Lighting – Algorithm Aided Lighting Design* (AUL) research project led by D.Sc. (Tech.) Aulikki Herneoja in 2011–2013. The AUL research team members are M.Sc. (Architect) Henrika Pihlajaniemi, M.Sc. (Architect) Toni Österlund, and research assistant, architecture student Tuulikki Tanska. The AUL project is mainly funded by the Academy of Finland.

Acconci Studio, 2011. Swarm Street. [online] Available at: ‹http://www.indyculturaltrail.org/media_release42811.html› [Accessed 31 August 2012].

Aish, R., 2013. First Build Your Tools. In: Peters, B. and Peters, T., eds., 2013. *Inside Smartgeometry:* Expanding the Architectural *Possibilities of Computational Design*. Chichester: John Wiley & Sons Ltd, pp. 36–49.

Autodesk, 2013a. *Maya*. [online] Available at: ‹http://www.autodesk.com/products/autodesk-maya/overview› [Accessed 1 August 2013].

Autodesk, 2013b. *3ds Max*. [online] Available at: ‹http://www.autodesk.com/products/autodesk-3ds-max/overview› [Accessed 1 August 2013].

Autodesk, 2013c. *DesignScript*. [online] Available at: ‹http://labs.autodesk.com/utilities/designscript› [Accessed 1 August 2013].

Bandini, S., Manzoni, S. and Simone, C., 2002. Dealing with Space in Multi-Agent Systems: a model for Situated MAS. In: *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3 (AAMAS '02)*, pp. 1183–1190.

Bandini, S., Federici, M.L. and Vizzari G., 2007. Situated Cellular Agents Approach to Crowd Modelling and Simulation. *Cybernetics and Systems*, 38 (7), pp. 729–753.

Bandini, S., Bonomi A., Vizzari, G. and Acconci, V., 2009a. An Asynchronous Cellular Automata-Based Adaptive Illumination Facility. In: R. Serra, R. Cucchiara eds. 2009, *AI*IA 2009: Emergent Perspectives in Artificial Intelligence*, LNAI 5883, Berlin Heidelberg: Springer-Verlag, pp. 405–415.

Bandini, S., Bonomi, A., Vizzari, G. and Acconci, V., 2009b. A CA-Based Self-Organising Environment: A Configurable Adaptive Illumination Facility. In: V. Malyshkin, ed., 2009, *PaCt 2009*, LNCS 5698, Berlin Heidelberg: Springer-Verlag, pp. 153–167.

Bandini, S., Bonomi, A., Vizzari, G. and Acconci, V., 2010. A Cellular Automata-Based Modular Lighting System. In: S. Bandini et al., eds., 2010, *ACRI 2010*, LNCS 6350, Berlin Heidelberg: Springer-Verlag, pp. 334–344.

Bertalanffy, L. von, 1969. *General System Theory: Foundations, Development, Applications*. New York: George Braziller.

Blender Foundation, 2013. Blender. [online] Available at: http://www.blender.org/› [Accessed 1 August 2013].

Borries, F. von, Walz, S. P. and Böttger, M. eds. 2007. *Space Time Play: Computer Games, Architecture and Urbanism: the Next Level*. Basel: Birkhäuser.

Cinder, 2013. [online] Available at: ‹http://libcinder.org/› [Accessed 1 August 2013].

Coates, P., 2010. *programming.architecture*. Abingdon: Routledge.

Essen, H. van, Offermans, S. and Eggen, B., 2012. Exploring the Role of Autonomous System Behaviour in Lighting Control. In: DIS 2012, *workshop on Designing Interactive Lighting*, June 11, 2012: Newcastle, UK.

Frazer, J., 1995. *An Evolutionary Architecture*. London: Architecture Association.

Fricker, P., Wartmann C., Hovestadt, L., 2008. Processing: Programming instead of Drawing. In: *Proceedings of eCAADe 2008*, Antwerp, Belgium, pp. 525–530.

Hansmeyer, M., 2003. L-Systems in Architecture. [online] Available at: ‹http://www.michael-hansmeyer.com/flash/l-systems.html› [Accessed 31 August 2012].

Heer J., Bostock M. and Ogievetsky V., 2010. A tour through the visualisation zoo. *Communications of the ACM*, 53 (6), pp. 59–67.

Hemberg, M., 2009. GENR8 A surface design tool. [online] Available at: ‹http://projects.csail.mit.edu/emergentDesign/genr8/› [Accessed 31 August 2012]

Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems*. Cambridge, MA: The MIT Press.

Holland, J.H., 1998. *Emergence: From Chaos to Order*. Reading, Berkshire: Oxford University Press.

Jennings, N., Sycara, K. and Wooldridge, M., 1998. A Roadmap of Agent Research and Development. In: *Autonomous Agents and Multi-Agent Systems*, 1 (1), pp.7–38.

Kauffman, S., 1993. *The Origins of Order: Self-organization and Selection in Evolution*. New York: Oxford University Press.

Kauffman, S., 1995. *At Home in the Universe: The search for Laws of Complexity*. New York: Oxford University Press.

Lighting Design Collective, 2011. Silo 468. [online] Available at: ‹http://www.ldcol.com/blog/#silo468› [Accessed 30 August 2012].

Macal, C. M., North, M. J., 2009. Agent-based Modeling and Simulation. In: Rossetti, M. D., Hill, R. R., Dunkin, A. and Ingalls, R. G., eds., 2009. *Proceedings of the 2009 Winter Simulation Conference – Wsc 2009*, December 13–16, 2009, Austin, TX, USA, pp.86–98.

McCullough, M, 2004. *Digital Ground*. Cambridge: Polity Press.

McNeel, 2013. *Rhinoceros*. [online] Available at: < http://www.rhino3d.com/ > [Accessed 1 August 2013].

Menges, A. and Ahlquist, S. eds., 2011. *Computational Design Thinking*. Padstow, Cornwall: Johan Wiley & Sons Ltd.

Mitchell, M., 2009. *Complexity: A Guided Tour*. New York: Oxford University Press.

Mitchell, W. J., 1990. *The Logic of Architecture: Design, Computation and Cognition*. Cambridge, MA: The MIT Press.

Niazi, M. and Hussain, A., 2011. Agent-based computing from multi-agent systems to agent-based Models: a visual survey. In: *Scientometrics*, 89 (2), pp. 479–499.

North, M. J., Collier, N. T., Ozik, J., Tatara, E., Altaweel, M., Macal, C. M., Bragen, M. and Sydelko P., 2013. *Complex Adaptive Systems Modeling with Repast Simphony*. Complex Adaptive Systems Modeling 2013, 1 (3).

OpenFrameworks, 2013. [online] Available at: ‹http://www.openframeworks.cc/› [Accessed 1 August 2013].

O'Sullivan, D. B., 2000. *Graph-based cellular automaton models of urban spatial processes*. Ph.D. University College London. Available at: ‹http://web.env.auckland.ac.nz/people_profiles/osullivan_d/personal/phd/index.html› [Accessed 30 August 2012].

Partanen, J. and Joutsiniemi, A., 2007. Simulaatio kaupungin kompleksisen kehityksen hallinnassa [Simulation in the control of the complex development of city]. *Yhdyskuntasuunnittelu*, 45 (2), pp. 44–63.

Peters, B., and De Kestelier, X., 2013. Computation works: The Building of Algorithmic Thought. *Architectural Design*, 83 (2), pp.8–15.

Pihlajaniemi, H., Österlund, T. and Herneoja, A., 2011. Adaptive urban lighting: inhabitant and the sense of control?. *Inhabiting Adaptive Architecture. Workshop at the International Conference on Adaptive Architecture 2011*, London, UK, 3–5 March 2011, unpublished.

Pihlajaniemi, H., Luusua, A., Österlund, T. and Tanska, T., 2012. LightStories: Designing and Evaluating Interactive and Participatory Urban Lighting. In: DIS 2012, *workshop on Designing Interactive Lighting*, June 11, 2012: Newcastle UK.

Pihlajaniemi, H., Österlund, T. and Tanska, T., 2012. Tools for interaction and user participation in urban lighting; LightStories case. In: H. Achten, J. Pavlicek, J. Hulin, D. Matejdan, eds. 2012. *Physical Digitality – Proceedings of the 30th eCAADe Conference – Volume2*, Prague: Czech Technical University in Prague, Faculty of Architecture, pp. 677–687.

Poulsen, E.S., Andersen, H.J. and Jensen, O.B., 2012. Full Scale Experiment with Interactive Urban Lighting. In: DIS 2012, *workshop on Designing Interactive Lighting*, June 11, 2012: Newcastle, UK.

Prusinkiewicz, P. and Lindenmayer, A., 1990. *The algorithmic beauty of plants*. New York: Springer-Verlag.

Prusinkiewicz, P., 2000. Paradigms of pattern formation: Towards a computational theory of morphogenesis. In: Carbone, A., Gromov, M. and Prusinkiewicz, P. eds., 2000. *Pattern Formation in Biology, Vision, and Dynamics*, River Edge, NJ: World Scientific, pp. 91–95.

rAndom International, 2010. *Swarm Light*. [online] Available at: ‹http://random-international.com/work/swarm-light/› [Accessed 31 August 2012].

Reas, C. and Fry, B., 2007. *Processing: A Programming Handbook for Visual Designers and Artists*. Boston: MIT Press.

Resnick, M., 1994. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: The MIT Press.

Reynolds, C. W., 1987. Flocks, herds and schools: A distributed behavioral model. *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, ACM 21* (4), pp. 25–34.

Reynolds, C. W., 1999. Steering Behaviors For Autonomous Characters. In: Yu, A. ed., 1999. *Proceedings of the 1999 Game Developer's Conference*, San Francisco: Miller Freeman, pp. 763–782.

Seitinger, S., 2010. *Liberated Pixels: Alternative Narratives for Lighting Future Cities*. Ph.D. Princeton University.

Shadbolt, N., 2003. Ambient Intelligence. *IEEE Intelligent Systems*, 18 (4), pp.2–3.

Stiny, G., 1980. Introduction to shape and shape grammars. *Environment and Planning B*, 7 (3), pp. 343–351.

Studio Drift, 2011. *Flylight*. [online] Available at: ‹http://www.designdrift.nl/work/flylight-2› [Accessed 30 August 2012].

Terzidis, K., 2006. *Algorithmic Architecture*. Amsterdam: Elsevier.

Thomsen, B. S., 2009. *Performative Environments*. Ph.D. Aalborg University. Available at: ‹http://vbn.aau.dk/files/19929583/Bo_Stjerne_PhD_Performative_Environments_distribute.pdf› [Accessed 1 August 2013].

Thomsen, B. S. and Poulsen, E. S., 2010. Performative Architecture and Street Scapes. In: Kiib, H. ed., 2010. *Performative Urban Design*. Aarhus: Aalborg Universitetsforlag, pp. 78–91.

Villareal, L., 2009. *Multiverse*. [online] Available at: ‹http://www.villareal.net/project_1.html› [Accessed 31 August 2012].

Vrachliotis, G., 2007. Game of Life: On Architecture, Complexity and the Concept of Nature as a Game. In: Borries, F. von, Walz, S. P. and Böttger, M. eds., 2007. *Space Time Play: Computer Games, Architecture and Urbanism: the Next Level*. Basel: Birkhäuser, pp. 340–343.

vvvv, 2013. [online] Available at: ‹http://vvvv.org/› [Accessed 1 August 2013].

Waldrop, M. M., 1992. *Complexity*. New York: Simon & Schuster.

Weinstock, M., 2010. *The Architecture of Emergence: The Evolution of Form in Nature and Civilization*. Chichester: John Wiley & Sons.
Weiser, M., 1991. The Computer for the 21$^{st}$ Century. *Scientific American*, 265 (3), pp. 94–104.

Wilensky, U., 1999. *NetLogo*. [online] Evanston, IL.: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Available at: ‹http//ccl.northwestern.edu/netlogo/› [Accessed 1 August 2013].

Wolfram, S., 2002. *A New Kind of Science*. Champaign, IL: Wolfram Media.

Wurzer, G., Alaçam, S., Lorenz, W., 2011. How to Teach Architects (Computer) Programming: A Case Study. In: *Proceedings of the 29$^{th}$ eCAADe Conference*, Slovenia: University of Ljubljana, Faculty of Architecture, pp. 51–56.

Yamins, D., Nagpal, R., 2008. Automated global-to-local programming in 1-d spatial multi-agent systems. In: Padgham, Parkes, Müller and Parsons eds. *Proc. of 7$^{th}$ Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS2008)*, 12–16. May, Estoril, Portugal, pp. 615–622.

## Biographical information

Toni Österlund,
M.Sc. Architect, researcher
Place of work:
Adaptive Urban Lighting project
Department of Architecture
Address:
P.O. Box 4100, FI90014,
University of Oulu, Finland
Telephone number:
+358 40 5086780
E-mail: toni.osterlund@oulu.fi

Toni Österlund is an architect, currently doing his Ph.D. in the *Adaptive Urban Lighting – Algorithm Aided Lighting Design* research project at the University Of Oulu, Department of architecture, Finland, with the working thesis title *Emergent Algorithms for Adaptive Urban Lighting – Emergence in Adaptation of Urban Lighting in Pedestrian Context*. Toni Österlund is interested with the possibilities of digital fabrication, algorithm aided design methods and implementing natural processes in architecture.