Supporting the Design of Self-Organizing Ambient Intelligent Systems Through Agent–Based Simulation

Stefania Bandini, Andrea Bonomi, Giuseppe Vizzari
Complex Systems and Artificial Intelligence research center, University of Milano–Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
{stefania.bandini,andrea.bonomi,giuseppe.vizzari}@csai.disco.unimib.it

Abstract— **The ambient intelligence scenario depicts electronic environments that are sensitive and responsive to the presence of people. The aims of this kind of system is not necessarily to provide some form of electronic service to its users, but also to enhance the everyday experience of people moving inside the related physical environment. For this second type of application, computer simulation represents a useful way to envision the behaviour of responsive environments without actually bringing them into existence in the real world. This paper will describe the simulation of an adaptive illumination facility, a physical environment endowed with a set of sensors that perceive the presence of humans (or other entities such as dogs, bicycles, cars) and interact with a set of actuators (lights) that coordinate their state to adapt the ambient illumination to the presence and behaviours of its users. This system is made up of a model managing the self-organization of the adaptive illumination facility and an agent–based model to simulate pedestrian dynamics in the physical environment in which the system is deployed.**

I. INTRODUCTION

The ambient intelligence scenario [14] depicts future human environments endowed with a large number of electronic devices, interconnected by means of wireless communication facilities, able to perceive and react to the presence of people.

The goals of these facilities can be very different, from providing electronic services to humans accessing these environment through computational devices (such as personal computers or PDAs), to simply providing some form of ambient adaptation to the users' presence (or voice, or gestures), without thus requiring him/her to employ a computational device. Ambient intelligence comprises thus those systems that are designed to autonomously adapt the environment to the people living or simply passing by in it in order to improve their everyday experience.

Besides the specific aims of the ambient intelligent system, there is an increasing interest and number of research efforts on approaches, models and mechanisms supporting forms of self-organization and management of the components (both hardware and software) of such systems. The latter are growingly viewed in terms of autonomous entities, managing internal resources and interacting with surrounding ones so as to obtain the desired overall system behaviour as a result of local actions and interactions among system components. Examples of this kind of approach can be found in both in relatively traditional pervasive computing applications (see, e.g., [8]), but also in a new wave of systems developed in the vein of amorphous computing [2] such as the one on paintable computers described in [7]. In this rather extreme application a whole display architecture is composed of autonomous and interacting graphic systems, each devoted to a single pixel, that must thus interact and coordinate their behaviours even to display a simple character.

Computer simulation plays an important role in supporting the design and realization of adaptive, self-organizing ambient intelligence systems. In fact, traditional design and modeling instruments can provide a suitable support for evaluating static properties of this kind of environment (e.g. through the construction of 3D models representing a mock-up, proof of concept of the desired appearance or also adaptation effect but in a single specific situation), but they are not designed to provide abstractions and mechanisms for the definition and simulation of reactive environments and their behaviours. Through specific models and simulators it is possible to obtain an envisioning of the static features of the ambient intelligence system as well as its dynamic response to the behaviour of humans and other relevant entities situated in it. This allows performing a *face validation* [13] of the adaptation mechanisms and also to perform a tuning of the relevant parameters.

This paper describes the application of a modeling and simulation approach to support the design of an adaptive illumination facility that is being designed and realized by the Acconci Studio[1] in Indianapolis. In particular, the designed system should be able to locally enhance the overall illumination of a tunnel in order to highlight the position and close surrounding area of pedestrians (as well as other entities such as dogs, bicycles, cars). In this case, the simulation offers both a support to the decisions about the number and positioning of lights and, more important, it encapsulates the self-organization mechanisms guiding the adaptive behaviour of lights reacting the the presence of pedestrians and other relevant entities in the environment. By providing the current state of the environment, in terms of simulated outputs of sensors detecting the presence of pedestrians, as an input to the self-organization model it is possible to obtain its simulated response, and the current state of lights. A schema

---

[1] http://www.acconci.com

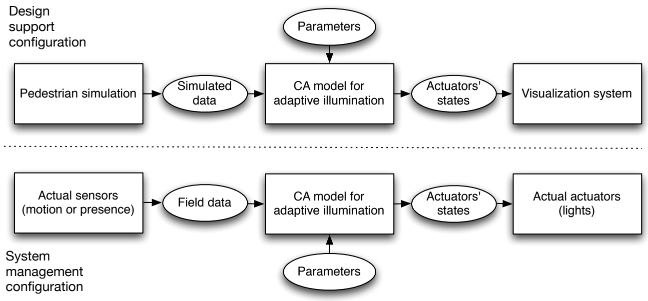


Fig. 1. A schema describing the modules of the design support system prototype.

of the overall simulation system is shown in Figure 1: it must be noted that the self-organization model adopted for the simulator could be effectively used to manage the actual system, simply providing actual inputs from field sensors and employing its outputs to manage actual lights rather that a virtual visualization of the actual environment.

The following section will introduce more in details the specific scenario in which this research effort is set, describing the requirements for the adaptive illumination system and the environment adaptation model. Section IV introduces the pedestrian modeling approach, while the self-orga-nization model guiding the adaptive illumination facility is described in Section III. A description of the developed environment supporting designers will follow, then conclusions and future works will end the paper.

II. THE SCENARIO

The Acconci Studio, partner of the described research effort, has recently been involved in a project for the renovation of a tunnel in the Viginia Avenue Garage in Indianapolis. The tunnel is currently mostly devoted to cars, with relatively limited space on the sidewalks and its illumination is strictly functional. The planned renovation for the tunnel includes a set of interventions, and in particular two main effects of illumination, also depicted in a graphical elaboration of the desired visual effect shown in Figure 2: an overall effect of *uniformly coloring* the environment through a background, ambient light that can change through time, but slowly with respect to the movements and immediate perceptions of people passing in the tunnel; a *local effect of illumination* reacting to the presence of pedestrians, bicycles, cars and other physical entities.

The rationale of this local and dynamic adaptive illumination effect is better explained by the following narrative description of the desired effect:

> The color is there to make a heaviness, a thickness, only so that the thickness can be broken. The thickness is pierced through with something, theres a sparkle, its you that sparkles, walking or cycling though the passage, this tunnel of color. Well no, not really, its not you: but its you that sets off the sparkle a sparkle here, sparkle there, then another sparkle in-between one sparkle affects the other, pulls the other, like a magnet a point of sparkle is stretched out into a line of sparkles is stretched out into a network of sparkles.
>
> These sparkles are above you, below you, they spread out in front of you, they light your way through the tunnel. The sparkles multiply: its you who sets them off, only you, but – when another person comes toward you in the opposite direction, when another person passes you, when a car passes by some of these sparkles, some of these fire-flies, have found a new attractor, they go off in a different direction.

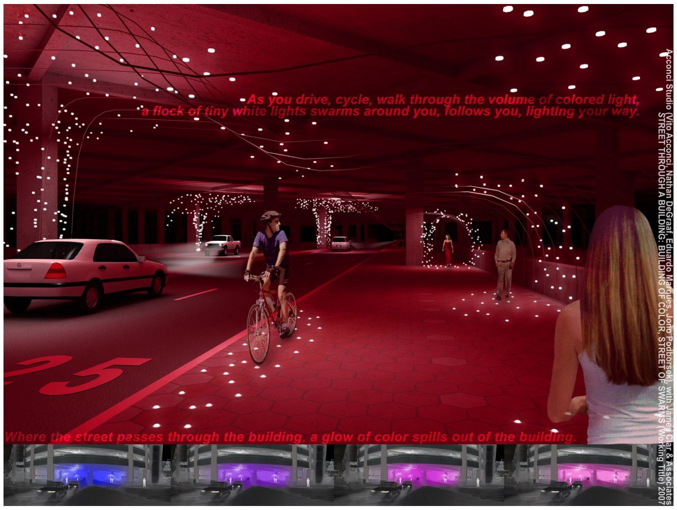


Fig. 2. A visual elaboration of the desired adaptive illumination facility (the image appears courtesy of the Acconci Studio).

The first type of effect can be achieved in a relatively simple and centralized way, requiring in fact a uniform type of illumination that has a slow dynamic. The second point requires a different view on the illumination facility. In particular, it must be able to perceive the presence of pedestrians and other physical entities passing in it, in other words it must be endowed with sensors (detecting either the presence or the movement of relatively big objects). Moreover, it must be able to exhibit local changes as a reaction to the outputs of the aforementioned sensors, providing thus for a non uniform component to the overall illumination. The overall environment must be thus split into parts, cells that represent proper subsystems: Figure 3 shows a schema of the approach we adopted to subdivide the physical environment into autonomous units, provided with motion/presence sensors (able to detect the arrival/presence of relevant entities) and lights (to adapt the ambient illumination, highlighting the presence of pedestrians).

However, the effect of the presence of a pedestrian in a portion of space should extend beyond the borders of the occupied cell. In fact, the illumination effect should "light the way" of a pedestrian through the tunnel. Cells must thus be able to interact, in order to influence neighboring ones whenever a pedestrian is detected, to trigger a (maybe less intense) illumination. The model we adopted to manage this form of self-organization of the illumination facility is a Cellular Automata (CA) [15], whose transition rule defines and

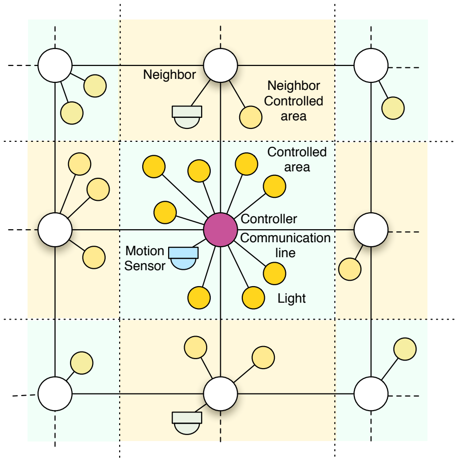


Fig. 3. A schema of the CA model for the adaptive illumination facility.

manages the interaction among cells and thus the influence of the presence of a pedestrian on neighboring ones.

III. Adaptive Illumination Model

We employed a Cellular Automata model to realize the local effect of illumination as a self-organized reaction to the presence of pedestrians. CA cells, related to a portion of the physical environment, comprise sensors and actuators, as schematized in Figure 3. The former can trigger the behaviours of the latter, both through the interaction of elements enclosed in the same cell and by means of the local interaction among adjacent cells. The transition rule models mechanisms of reaction and diffusion, and it was derived by previous applications to reproduce natural phenomena such as percolation processes of pesticides in the soil, in percolation beds for the coffee industry and for the experimentation of elasticity properties of batches for tires [3]. In this specific application the rule manages the interactions of cells arranged through a multilayered architecture based on the Multilayered Automata Network model [6], schematized in Figure 3.

Multilayered Automata Network have been defined as a generalization of Automata Networks [10]. The main feature of the Multilayered Automata Network is the explicit introduction of a hierarchical structure based on nested graphs, that are graphs whose vertexes can be in turn be a nested graph of lower level. A Multilayered Automata Network is directly obtained from the nested graph structure by introducing states and a transition function.

The irregular nature of the cellular space is not the only difference between the adopted approach and the traditional CA models. In fact, CAs are in general closed and synchronous systems, in which cells update their state in parallel triggered by a global clock. Dissipative Cellular Automata (DCA) [16] differ from the basic CAs mainly for two characteristics: while CA are synchronous and closed systems, DCA are open and asynchronous. DCA cells are characterized by a thread of control of their own, autonomously managing the elaboration of the local cell state transition rule. DCA can thus be considered as an open agent system [12], in witch the cells update their state independently of each other and they are directly influenced by the environment.

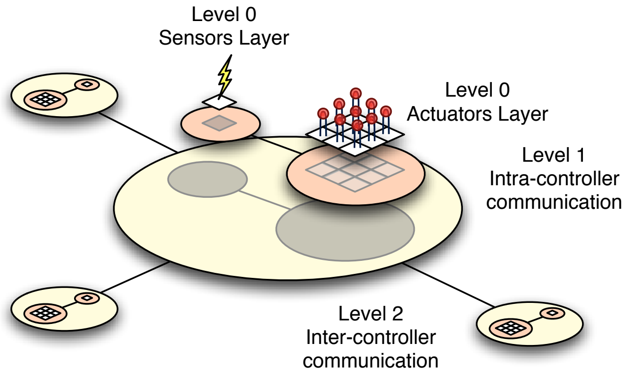


Fig. 4. The proposed automata network for the D-MAN.

The model we defined and adopted, Dissipative Multilayered Automata Network (D-MAN), takes thus the advantages of both the Multilayered Automata Network and the Dissipative Cellular Automata. An informal definition this model describes D-MAN as Multilayered Automata Network in which the cells update their state in an asynchronous way and they are open to influences by the external environment.

The multilayered cellular structure of the D-MAN is composed of three layers: the first level is related to the basic *discretization* of the physical environment into cells, corresponding to a local controller. Each of these cells comprises two additional levels, respectively devoted to the *perception* and *actuation* responsibilities of the higher level cell. This structure is schematized in Figure 4. Specific transition rules must thus be defined to manage different interactions and influences that take place in this structure, and mainly (i) the direct influence of a sensor that detected a pedestrian to the actuators in the same cell, and (ii) the influence of a high level cell to the neighboring ones (given the internal structure of each cell, due to the presence of a specific level of actuators inside it, this interaction effectively affects *a part* of a neighboring cell). Moreover, the effect of external stimuli must gradually vanish, and lights must fade in absence of pedestrians.

The adaptive illumination model is thus characterized by several features that make it difficult to predict how it will react to particular stimuli (i.e. patterns of pedestrian movement in the related environment), from the number and positioning of sensors and actuators, to the parameters of the transition rule. To couple this model with a pedestrian simulation model sharing the discrete representation of the spatial aspect of the environment allows to simulate the behaviour of the adaptive illumination facility as a response to specific patterns of usage of the environment by pedestrians.

A. Model Architecture

The designed system is an homogeneous peer system. As shown in Figure 3, every controller has the responsibility of managing the sensors and actuators belonging to a fixed area of the space. Controllers are homogeneous in terms of hardware and software capabilities. Every controller is connected to a motion sensor, which roughly covers the controlled area, some lights (about 40 LED lights) and neighbouring controllers.

As shown in Figure 4, the external layer (level 2) is the communication layer between the controllers of the system. Every controller is an automata network of two nodes, one node is a sensor communication layer and represents a space in which every sensor connected to the microcontroller has a correspondent cell. The other node represents the actuators' layer in which the cells pilot the actuators (lights, in our case). Since the external layer is a physical one and every cell is an independent microcontroller, it cannot be assumed that the entrire network is synchronized. In same cases, a synchronous network can be constructed (for example, a single clock devices can be connected to each microcontrollers or the microcontrollers can be synchronized by a process without a master node), but the most general case is an asynchronous network.

B. Sensors Layer

The Sensor Layer is a Level 0 Dissipative Automata. As previously introduced, it is composed of a single cell, since only one sensor is connected to each microcontroller. It is a Dissipative Automata because the internal state of the cell is influenced by the external environment. The state of the cell is represented by a single numerical value $v_s \in \mathbb{N}_{8bit}$, where

$$\mathbb{N}_{8bit} \subset \mathbb{N}_0, \forall x : x \in \mathbb{N}_{8bit} \Rightarrow x < 2^8$$

The limit value was chosen for performance reasons because 8-bit microcontrollers are widely diffused and they can be sufficiently powerful to manage this kind of situation. The value of v_s is computed as

$$v_s(t+1) = v_s(t) \cdot m + s(t+1) \cdot (1-m)$$

where $m \in \mathbb{R}, 0 \leq m \leq 1$ is the *memory coefficient* that indicates the degree of correlation between the previous value of v_s and the new value, $s(t) \in N_{8bit}$ is the reading of the sensor at the time *s(t)*. If the sensor is capable of distance measuring, *s(t)* is inverse proportional to the measured distance (so, if the distance is 0, the value is 255, if the distance is ∞ the value is 0). If the sensor is a motion detector sensor (it able to signal 1 if an object is present or 0 otherwise) *s(t)*, s(t) is equal to 0 if there is not detected motion, *c* in case of motion, where $c \in \mathbb{N}_{8bit}$ is a constant (in our tests, 128 and 192 are good values for *c*).

C. Diffusion Rule

In this section we describe the diffusion rule, that is used to propagate the sensors signals through the system. At a given time, every level 2 cell is characterized by an intensity of the signal, $v \in \mathbb{N}_{8bit}$. Informally, the value of v at time $t+1$ depends of the value of v at time t and on the value of $v_s(t+1)$, to capture both the aspects of interaction with neighbouring cells and the memory of the previous external stimulus caused by the presence of a physical entity in the area associated to the cell.

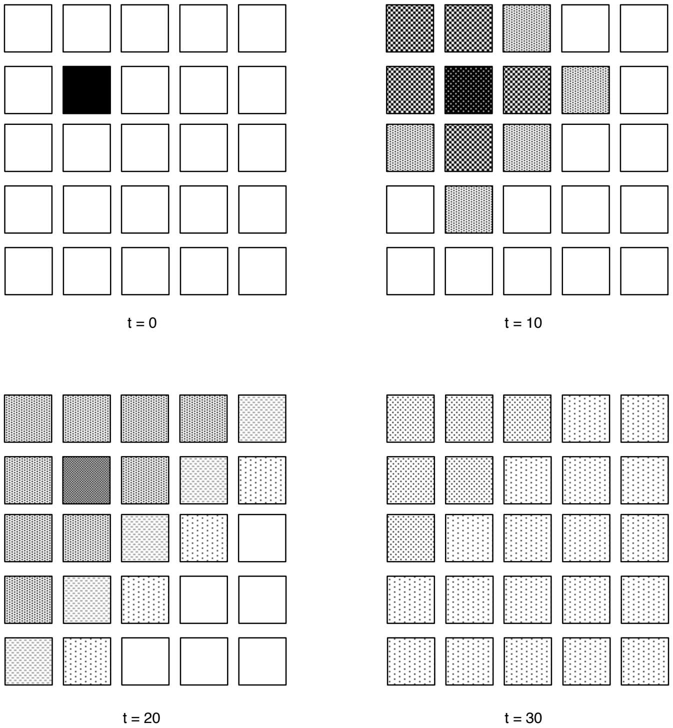


Fig. 5. An example of the dynamic behaviour of a diffusion operation. The signal intensity is spread throughout the lattice, leading to a uniform value; the total signal intensity remains stable through time, since evaporation was not considered.

The intensity of the signal decreases over time, in a process we call evaporation. In particular, let us define $\epsilon_{evp}(v)$ as the function that computes the quantity of signal to decrement from the signal and is defined as

$$\epsilon_{evp}(v) = v \cdot e_1 + e_0$$

where $e_0 \in \mathbb{R}^+$ is a constant evaporation quantity and $e_1 \in \mathbb{R}, 0 \leq e_1 \leq 1$ is the evaporation rate (e.g. a value of 0.1 means a 10% evaporation rate).

The evaporation function $evp(v)$, computing the intensity of signal v from time t to $t+1$, is thus defined as

$$evp(v) = \begin{cases} 0 & \text{if } \epsilon_{evp}(v) > v \\ v - \epsilon_{evp}(v) & \text{otherwise} \end{cases}$$

The evaporation function is used in combination with the neighbours' signal intensities to compute the new intensity of a given cell. We first present the formula for a regular neighbourhood and than we generalize to the irregular structure.

1) Regular neighbourhood: The automaton is contained in the finite two-dimensional square grid $\mathbb{N}^2$. We suppose that the cell $C_{i,j}$ is located on the grid at the position *i, j*, where $i \in \mathbb{N}$ and $j \in \mathbb{N}$. According to the von Neumann neighbourood [11], a cell $C_{i,j}$ (unless it is placed on the border of the lattice) has 4 neighbours (as shown in figure 6), denoted by $C_{i-1,j}$, $C_{i,j+1}$, $C_{i+1,j}$, $C_{i,j-1}$.

For simplicity, we numbered the neighbours of a cell from 1 to 4, so for the cell $C_{i,j}$, N_1 is $C_{i-1,j}$, N_2 is $C_{i,j+1}$, N_3 is $C_{i+1,j}$, and N_4 is $C_{i,j-1}$

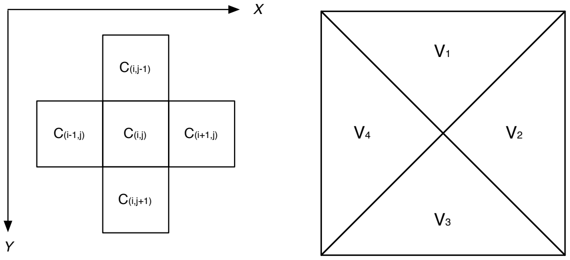


Fig. 6. On the left, the von Neumann neighbourhood of the cell $C_{i,j}$, on the right, the internal structure of a cell of the regular automaton.

At a given time, every cell is characterize by an intensity of the sensor signal. Each cell is divided into four parts (as shown in Figure 6), each part can have a different signal intensity, and the overall intensity of the signal of the cell is the sum of the parts intensity values. The state of each cell $C_{i,j}$ of the automaton is defined by $C_{i,j} = \langle v_1, v_2, v_3, v_4 \rangle$ where $v_1, v_2, v_3, v_4 \in \mathbb{N}_{8bit}$ represent the intensity of the signal of the 4 subparts. $V_{i,j}(t)$ represents the total intensity of the signals (i.e. the sum of the subparts signal intensity) of the cell i, j at time t. The total intensity of the neighbours are denoted by V_{N1}, V_{N2}, V_{N3}, and V_{N4}. The signal intensity of the subparts and the total intensity are computed with the following formulas:

$$v_j(t+1) = \begin{cases} \frac{evp(V(t)) \cdot q + evp(V_{Nj}(t)) \cdot (1-q)}{4} & \text{if } \exists N_j \\ \frac{evp(V(t))}{4} & \text{otherwise} \end{cases}$$

$$V(t+1) = \sum_{i=1}^{4} v_i(t+1)$$

where $q \in \mathbb{R}, 0 \le q \le 1$ is the conservation coefficient (i.e. if q is equals to 0, the new state of a cell is not influenced by the neighbours values, if it is equals to 0.5 the new values is a mean among the previous value of the cell an the neighbours value, if it is equals to 1, the new value does not depend on the previous value of the cell but only from the neighbours). The effect of this modeling choice is that the parts of cells along the border of the lattice are only influenced through time by the contributions of other parts (that are adjacent to inner cells of the lattice) to the overall cell intensity.

2) Irregular neighbourhood: The irregular structure automata is a generalization of the regular one. The automaton is composed of cell numbered from 1 to N, so we use C_i for $0 \le i \le N$ to indicate the i-th cell. Every cell C_i can have an arbitrary number of neighbours $L_i, 0 \le L_i \le L \le N-1$ where L_i is the numbers of neighbours of the cell C_i and $L = \max(L_i)$ is the maximum numbers of neighbours of every cell the system. Neighbouring cells of cell i can be denoted as $N_{i,l}$.

As for the regular neighbourhood case, each cell is divided into L parts, each part can have a different signal intensity, and the overall intensity of the signal of the cell is the sum of the parts intensity values. The state of each cell C_i of the automaton is defined as $V_i = \sum_{l=0}^{L_i} v_{i,l}$ where $v_{i,l} \in \mathbb{N}_{8bit}$ represent the intensity of the signal of the L subparts. Finally, the intensity of each neighbouring cell of C_i is denoted by $V_{i,l}$.

The signal intensity of the subparts and the total intensity can thus be computed according to the following formulas:

$$v_{i,l}(t+1) = \begin{cases} \frac{evp(V_i(t)) \cdot q + evp(V_{i,l}(t)) \cdot (1-q)}{L} & \text{if } \exists N_{i,l} \\ \frac{evp(V_i(t))}{L_i} & \text{otherwise} \end{cases}$$

$$V_i(t+1) = \sum_{l=1}^{L_i} v_{i,l}(t+1)$$

In the real system, the maximum number of neighbours (L) is constrained by the number of available inputs on the microcontrollers.

D. Actuators Layer

The cells of the actuator layer determinate the actuators actions. In this project the actuators are LED lamps that are turned on and of according the the state of the cell. Instead of controlling a single LED from a cell, every cell is related to a group of LEDs disposed in the same (small) area.

In the case of regular neighbourhood, each controlled area in divided into 9 sub-areas and each sub-area contains a group of LEDs controlled by the same actuators layer cell. The state of each cell is influenced only by the state of the signal intensity of the upper layer cell. The correlation between the upper layer cell subparts and the actuators layer cells is shown in Figure 7.

The state of the actuators cells $A_1..A_9, A_j \in N_{8bit}$ is computed with the following formula:

$$A_i(t+1) = \begin{cases} v_i(t+1) & 1 \le i \le 4 \\ \dfrac{v_4(t+1) + v_1(t+1)}{2} & i = 5 \\ \dfrac{v_1(t+1) + v_2(t+1)}{2} & i = 6 \\ \dfrac{v_2(t+1) + v_3(t+1)}{2} & i = 7 \\ \dfrac{v_3(t+1) + v_4(t+1)}{2} & i = 8 \\ \dfrac{1}{4} \sum_{j=1}^{4} v_j(t+1) & i = 9 \end{cases}$$

There are differents approache to associtates the LEDs to the cells. A first approach consists to directly connected the lights intensity to the signal levels of the correspondent cell. Another approach consists to turn on a numbers of LEDs proportionals to the signal intensity of the controller cell.

IV. PEDESTRIAN SIMULATION MODEL

The adopted pedestrian model is based on the Situated Cellular Agent model, a specific class of Multilayered Multi-Agent Situated System (MMASS) [4] providing a single layered spatial structure for agents environment. A thorough description of the model is out of the scope of this paper, but we briefly introduce it to give some basic notion of the elements that are necessary to describe the SCA crowd modeling approach.

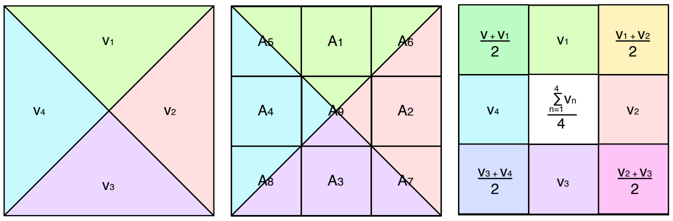


Fig. 7. Correlation between the upper layer cell subparts and the actuators layer cells.

A. Situated Cellular Agents

A *Situated Cellular Agent* system is defined by the triple $\langle Space, F, A\rangle$ where *Space* models the environment where the set A of agents is situated, acts autonomously and interacts through the propagation of the set F of fields and through reaction operations. *Space* consists of a set P of sites arranged in a network (i.e. an undirected graph of sites). The structure of the space can be represented as a neighborhood function, $N : P \rightarrow 2^P$ so that $N(p) \subseteq P$ is the set of sites adjacent to $p \in P$; the previously introduced *Space* element is thus the pair $\langle P, N\rangle$. Focusing instead on the single basic environmental elements, a site $p \in P$ can contain at most one agent and is defined by the 3–tuple $\langle a_p, F_p, P_p\rangle$ where:

- $a_p \in A \cup \{\perp\}$ is the agent situated in p ($a_p = \perp$ when no agent is situated in p that is, p is empty);
- $F_p \subset F$ is the set of fields active in p ($F_p = \emptyset$ when no field is active in p);
- $P_p \subset P$ is the set of sites adjacent to p (i.e. $N(p)$).

A SCA agent is defined by the 3–tuple $< s, p, \tau >$ where τ is the *agent type*, $s \in \Sigma_\tau$ denotes the *agent state* and can assume one of the values specified by its type (see below for Σ_τ definition), and $p \in P$ is the site of the $Space$ where the agent is situated. As previously stated, agent *type* is a specification of agent state, perceptive capabilities and behaviour. In fact an agent type τ is defined by the 3–tuple $\langle \Sigma_\tau, Perception_\tau, Action_\tau\rangle$. Σ_τ defines the set of states that agents of type τ can assume. $Perception_\tau : \Sigma_\tau \rightarrow [\mathbf{N} \times W_{f_1}] \ldots [\mathbf{N} \times W_{f_{|F|}}]$ is a function associating to each agent state a vector of pairs representing the *receptiveness coefficient* and *sensitivity thresholds* for that kind of field. $Action_\tau$ represents instead the behavioural specification for agents of type τ. Agent behaviour can be specified using a language that defines the following primitives:

- $emit(s, f, p)$: the $emit$ primitive allows an agent to *start the diffusion of field* f on p, that is the site it is placed on;
- $react(s, a_{p_1}, a_{p_2}, \ldots, a_{p_n}, s')$: this kind of primitive allows the specification of a *coordinated change of state* among adjacent agents. In order to preserve agents' autonomy, a compatible primitive must be included in the behavioural specification of all the involved agents; moreover when this coordination process takes place, every involved agents may dynamically decide to effectively agree to perform this operation;
- $transport(p, q)$: the $transport$ primitive allows the definition of *define agent movement* from site p to site q (that must be adjacent and vacant);
- $trigger(s, s')$: this primitive specifies that an agent must *change its state* when it senses a particular condition in its local context (i.e. its own site and the adjacent ones); this operation has the same effect of a reaction, but does not require a coordination with other agents.

For every primitive included in the behavioural specification of an agent type specific preconditions must be specified; moreover specific parameters must also be given (e.g. the specific field to be emitted in an emit primitive, or the conditions to identify the destination site in a transport) to precisely define the effect of the action, which was previously briefly described in general terms.

Each SCA agent is thus provided with a set of sensors that allows its interaction with the environment and other agents. At the same time, agents can constitute the source of given fields acting within a SCA space (e.g. noise emitted by a talking agent). Formally, a field type t is defined by $\langle W_t, \mathit{Diffusion}_t, Compare_t, Compose_t\rangle$ where W_t denotes the set of values that fields of type t can assume; $\mathit{Diffusion}_t : P \times W_f \times P \rightarrow (W_t)^+$ is the diffusion function of the field computing the value of a field on a given space site taking into account in which site (P is the set of sites that constitutes the SCA space) and with which value it has been generated. It must be noted that fields diffuse along the spatial structure of the environment, and more precisely a field diffuses from a source site to the ones that can be reached through arcs as long as its intensity is not voided by the diffusion function. $Compose_t : (W_t)^+ \rightarrow W_t$ expresses how fields of the same type have to be combined (for instance, in order to obtain the unique value of field type t at a site), and $Compare_t : W_t \times W_t \rightarrow \{True, False\}$ is the function that compares values of the same field type. This function is used in order to verify whether an agent can perceive a field value by comparing it with the sensitivity threshold after it has been modulated by the receptiveness coefficient.

B. SCA Based Pedestrian Model

The above introduced SCA model has beed applied to represent a very simple tunnel with two ends and some columns in it; pedestrians enter the tunnel from one end and they move towards the other end, avoiding obstacles either immobile (i.e. columns), and mobile (i.e. other pedestrians moving in the opposite direction).

The SCA *Space* is the same cellular space defined for the D-MAN described in Section III. To support agent navigation in this space, in each end of the tunnel we positioned an additional site in which a "beacon" agent (a static agent emitting a simple presence field) is situated. In the environment, thus, only two types of field are present.

To exploit this environmental specification in order to obtain the above overall system behaviour, we defined two types of agent, respectively interpreting the one type of field as attractive and ignoring the other one. This can be achieve through a simple $transport$ primitive, specifying that the agent should move towards the free adjacent site in which the intensity of the field considered attractive is maximum.

The behavioural specification of these agents is completed by an obstacle avoidance rule (another *transport* that moves the agent towards a random different lane whenever the best possible destination is occupied by an obstacle). Finally, agents reaching their destination, that is, one of the tunnel ends, are removed from the environment and they are positioned at the other end, so they start over their crossing of the tunnel.

V. THE DESIGN SUPPORT ENVIRONMENT

The design of human environments (e.g. buildings, stores, squares, roads) is a complex task, composed of several subtask evolving the initial idea into a detailed project, through the production of intermediate and increasingly detailed models.

After the initial phases, in which the designer usually expresses his/her creativity with sketches on the paper or on the computer, a Computer-Aided Design (CAD) software is used to develop the project in details. CAD softwares (e.g AutoCAD), and also 3D modelling applications (e.g. Autodesk 3DStudio Max, Blender) are used to create the digital models for the projects and to generate photo realistic renderings and animations. For a compact overview of the typical design process see [9].

Together with these software applications supporting designers in the definition of general architectural spaces, other tools supporting designers in very specific tasks can also be adopted: these tools vary from presenting the elaboration of the building shadows, to elaborating their impact on wind conditions, up to the simulation of vehicles and pedestrian movements in the designed scenario.

The proposed design environment is one of these tools; in particular, it helps the designers in the definition and specification of an adaptive illumination facility through the simulation of its dynamic behaviour. The output of the system is not only a graphical simulation but also a static configuration of the illumination facility (number of lights and their positioning) and an unambiguous specification of their dynamic behaviour (general lights self-organization model plus specific parameters).

The design environment is composed of two main modules: a simulation environment (that is in turn decomposed into a pedestrian simulation module and an adaptive illumination module) and a visualization facility. In the following paragraphs these modules will be described.

A. The Simulation Environment

The simulation environment actually comprises two models, one managing the network of controllers (with sensors and actuators), the other simulating the environment in which the adaptive illumination facility is situated and the pedestrians situated in it. The two simulations are connected: in particular, the state of sensors of is influenced by the state of the environment simulation.

The environment simulator, that is based on the MMASS [5], can be used to perform pedestrian simulation. This module actually feeds the self-organization model with simulated field data. The previously described CA model managing the self-organization of the illumination facility will react according to the current occupation of the space in the environment and according to its own parameters.

In this way, the designer can effectively envision the interaction between the people an the specified adaptive environment. The simulation environment allows the designer in configuring the network, defining the type, number, position of the sensors and actuators, and in specifying the behavior of the controllers, by means of defining the parameters of the CA model.

B. The Visualization Facility

The system supports both a 2D and 3D visualization of the simulated environment and the state of the two different enclosed models. The 2D visualization can be interactive, so it is possible to define an action event to be fired on a click (e.g. simulate the perception of a pedestrian when the designer clicks on a cell). This is useful because allows the designer to test the system behavior before specifying in an extensive way a pedestrian simulation scenario.

The 3D visualization is useful to understand the behavior of the system. It is not a photo realistic rendering but a real-time representation of simulated system. During the simulation, the user can navigate the 3D space, changing his/her point of view, for instance, taking the perspective of one of the pedestrians walking in the environment. It is possible to load 3D models both for the space and for pedestrian agents. The 3D visualization is based on the jMonkey engine[2]; the API of this open source project allows loading several 3D model formats. A screenshot of the visualization system is shown in Figure 8.

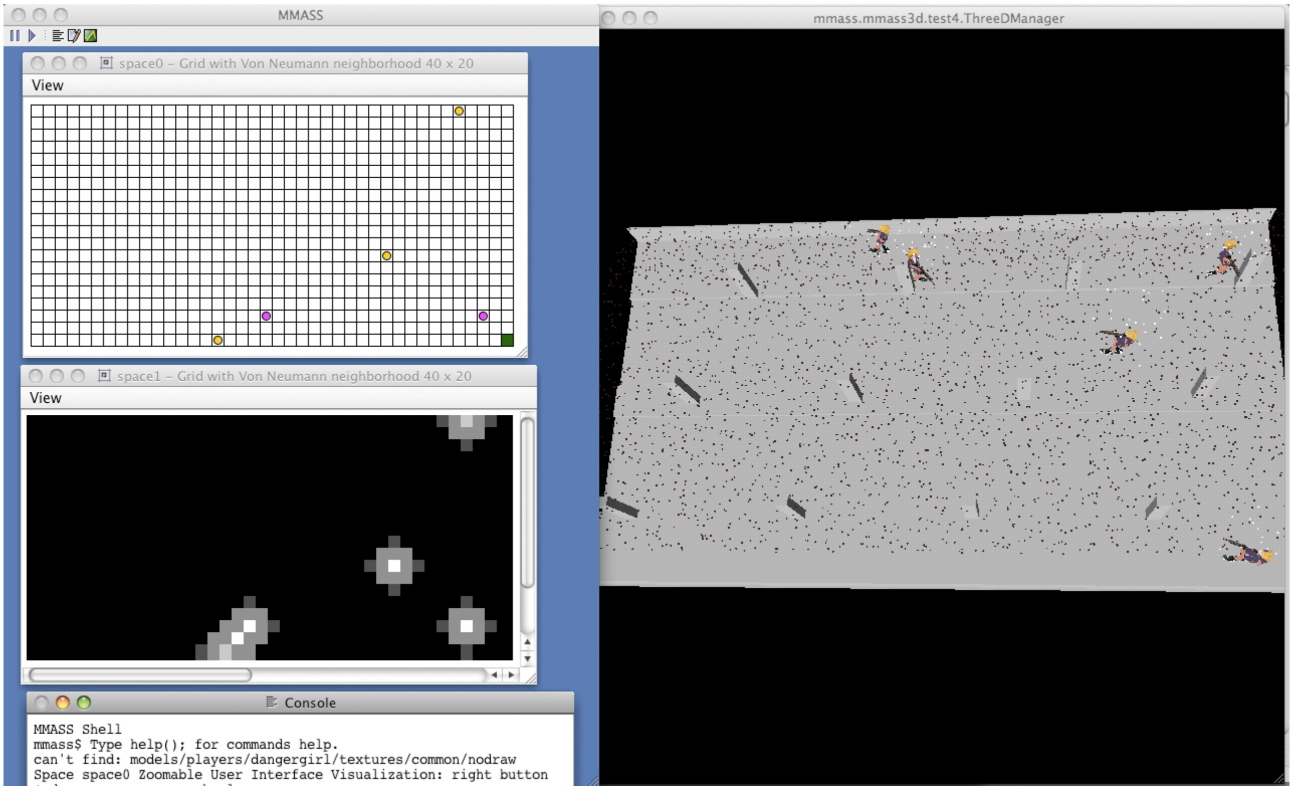
VI. FUTURE DEVELOPMENT

The paper introduced a simulation approach to supporting the design of an ambient intelligence infrastructure aimed at improving the everyday experience of pedestrians and people passing through the related environment. A specific scenario related to the definition and development of an adaptive illumination facility was introduced, and a CA-based model specifying its dynamic behaviour was defined. An agent-based pedestrian model simulating inputs and stimuli to the adaptation module was also introduced. A prototype of a system supporting designers in the definition of the relevant parameters for this model and for the overall illumination facility was finally described.

The renovation project is currently under development on the architectural and engineering side, whereas the CA-based model has shown its adequacy to the problem specification, both in order to provide a formal specification of the behaviour for the system components and possibly as a centralized control mechanism. The realized prototype explored the possibility of realizing an ad hoc tool that can integrate the traditional CAD systems for supporting designers in simulating and envisioning the dynamic behaviour of complex, self-organizating installations. It has been used to understand the adequacy of the modeling approach in reproducing the desired self-organized adaptive behaviour of the environment

---

[2] http://www.jmonkeyengine.com/



```
MMASS Shell
mmass$ Type help(); for commands help.
can't find: models/players/dangergirl/textures/common/nodraw
Space space0 Zoomable User Interface Visualization: right button
```

Fig. 8. Screenshot of the simulation environment: on the left, the top panel shows the position of pedestrians in the environment, while the bottom one shows the intensity of cells. The right panel shows a 3D visualization of the environment, including columns, lights and pedestrians.

to the presence of pedestrians. We are currently improving the prototype, on one hand, to provide a better support for the Indianapolis project and, on the other, to realize a more general framework for supporting designers of dynamic self-organizing environments.

ACKNOWLEDGEMENTS

The work presented in this paper has been partially funded by the University of Milano-Bicocca within the project “Fondo d’Ateneo per la Ricerca - anno 2007”.

REFERENCES

[1] *Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2007, Boston, MA, USA, July 9-11, 2007*. IEEE Computer Society, 2007.

[2] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. K. Jr., R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss. Amorphous computing. *Commun. ACM*, 43(5):74–82, 2000.

[3] S. Bandini, G. Erbacci, and G. Mauri. Implementing Cellular Automata Based Models on Parallel Architectures: The CAPP Project. In V. E. Malyshkin, editor, *PaCT*, volume 1662 of *Lecture Notes in Computer Science*, pages 167–179. Springer, 1999.

[4] S. Bandini, S. Manzoni, and C. Simone. 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*, pages 1183–1190. ACM Press, 2002.

[5] S. Bandini, S. Manzoni, and G. Vizzari. Towards a Platform for Multilayered Multi Agent Situated System Based Simulations: Focusing on Field Diffusion. *Applied Artificial Intelligence*, 20(4–5):327–351, 2006.

[6] S. Bandini and G. Mauri. Multilayered Cellular Automata. *Theor. Comput. Sci.*, 217(1):99–113, 1999.

[7] W. Butera. Text display and graphics control on a paintable computer. In *SASO* [1], pages 45–54.

[8] A. E. S. Filho, E. C. Lupu, N. Dulay, S. L. Keoh, K. P. Twidle, M. Sloman, S. Heeps, S. Strowes, and J. Sventek. Towards supporting interactions between self-managed cells. In *SASO* [1], pages 224–236.

[9] J. Frazer. Computing without computers. *Architectural Design*, 75(2):34–43, 2005.

[10] E. Goles and S. Martinez. *Neural and Automata Networks: Dynamical Behavior and Applications*. Kluwer Academic Publishers, 1990. ISBN 0-792-30632-5.

[11] H. Gutowitz. *Cellular Automata: Theory and Experiment*. MIT Press/Bradford Books, Cambridge Mass., 1991. ISBN 0-262-57086-6.

[12] N. R. Jennings. On agent-based software engineering. *Artif. Intell.*, 117(2):277–296, 2000.

[13] F. Klügl. A Validation Methodology for Agent-Based Simulations. In R. Menezes and M. Viroli, editors, *Symposium on Applied Computing*, pages 39–43. ACM Press, 2008.

[14] N. Shadbolt. Ambient Intelligence. *IEEE Intelligent Systems*, 18(4):2–3, 2003.

[15] J. von Neumann. *Theory of Self-Reproducting Automata*. University of Illinois Press, 1966.

[16] F. Zambonelli, M. Mamei, and A. Roli. What can cellular automata tell us about the behavior of large multi-agent systems? In A. F. Garcia, C. J. P. de Lucena, F. Zambonelli, A. Omicini, and J. Castro, editors, *SELMAS*, volume 2603 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 2002.