A Nature-inspired Approach for Large-Scale Pervasive Service Ecosystems

Cynthia Villalba and Franco Zambonelli

Department of Engineering Sciences and Methods
University of Modena and Reggio Emilia
Via Amendola 2, 42100 Reggio Emilia, Italy
{cynthia.villalba,franco.zambonelli}@unimore.it

Abstract. Innovative frameworks have to be identified for the deployment and execution of pervasive services made up of a massive number of components, and able to exhibit properties of *self-organization* and *self-adaptability*, and of *long-lasting evolvability*. This paper discuss how such frameworks should get inspiration from *Ecological systems*, modeling and deploying services as autonomous individuals (i.e *agents*), spatially-situated in an ecosystem of other services, data sources, and pervasive devices, all of which acting, interacting, and evolving according to a limited set of *"laws of nature"*. In this context, we present a reference architecture to frame the concepts and components of *eco-inspired* systems, discuss the characteristics of the ecological approach, and exemplify it with the help of a representative case study. Preliminary simulation results show the potential effectiveness of the approach.

1 Introduction

Pervasive and mobile computing devices increasingly populate our environments [5, 18]. These, together with the increasing amount of Web tools that make it possible to produce and access spatially-situated information about the physical world [3], will define a global-scale and very dense, decentralized infrastructure for general-purpose usage. At the user level, the infrastructure can be used to access innovative services for better perceiving/interacting with the physical world and for acting on it. It is also expected that users themselves will be able to personalize the infrastructure by deploying customized services over it. In addition, the infrastructure will be used as a way to enrich traditional classes of services with the capability of dynamically and autonomously adapting their behavior to the context in which they are exploited.

The effective development and execution of services in the above infrastructure calls for a deep rethinking of current service models and of service frameworks, in order to: (*i*) Naturally match the spatial nature of the environment and of the services within, and rely on mostly localized spatial interaction to provide support for massive scalability. (*ii*) Inherently exhibit properties of self-organization, self-adaptation and self-management that are required in highly-decentralized and highly-dynamic scenarios. (*iii*) Flexibly tolerate evolutions of structure and usage over time. This is necessary to account for increasingly diverse and demanding needs of users as well as for technological evolution, without forcing significant re-engineering to incorporate innovations and changes.

To reach this goal, we should no longer conceive services and their interactions as in usual service-oriented architectures [10], where services are simply functional entities orchestrated according to mostly static patterns and with the help of specific middleware services. No one can rely on ad-hoc one-of solutions to achieve features of self-* properties. Rather, the most promising direction is that of taking inspiration from natural systems [17, 9], where spatial concepts, self-organization, self-management, and long-lasting evolvability are inherently there because of the basic "rules of the game". We are aware that nature-inspired solutions have already been extensively exploited in the area of distributed computing for the implementation of specific middleware solutions or of specific distributed services [12]. Here we go further, by arguing that natural ecosystem can act as the key metaphor around which to conceive, model, and develop a fully-fledged pervasive service framework and all the components within.

Although one can think at different classes of natural systems and from different perspectives (e.g., physical, chemical, biological, or social) one can always recognize the following characteristics: above a spatial environmental substrate, autonomous individuals (i.e., agents) of different kinds interact, compete, and combine with each other in respect of the basic laws of nature. Accordingly, in our scenario, the shared pervasive infrastructure substrate will have to be conceived as the space in which bringing to life an ecosystem of service agents, intended as individuals whose computational activities are subject to some basic laws of the ecosystem, and for which the dynamics of the ecosystem (as determined by the enactment of its laws) will provide for naturally enforcing features of self-organization, self-management, and evolvability.

In this context, the paper provides the following contributions: at first, we introduce a reference architecture for nature-inspired pervasive service ecosystems, to show how ecosystem concepts can be framed into a unifying conceptual scheme, and briefly discuss the possible approaches to realize the architecture and the related works in the area (Section 2). We detail the specific ecological approach that we have started investigating. The approach abstracts the components of the ecosystem as sorts of "hungry" goal-oriented organisms that, driven by laws of survival, interact with each other and self-organize their activities according to dynamic food-web relations (Section 3). A simple yet representative case study is also introduced to present via test, performed in a simulation environment, a preliminary assessment of the potentials of our approach (Section 4). Section 5 concludes and outlines directions for future work.

2 Unified Architecture for Pervasive Service Ecosystems

A unifying reference architecture can be identified around which to frame the key abstractions and the conceptual structure for spatial pervasive service ecosystems, independently of the specific metaphor adopted (see Figure 1).

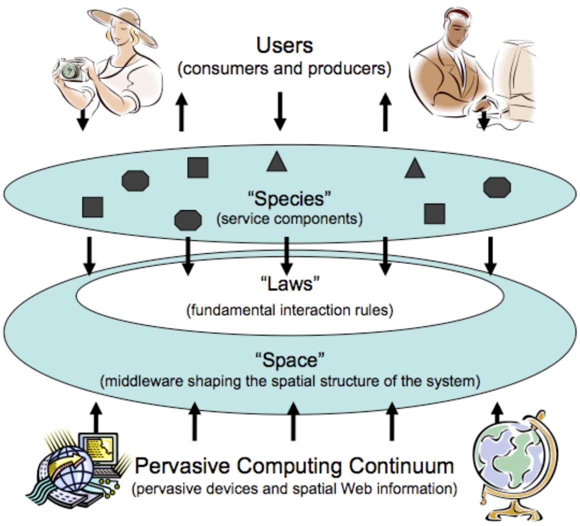

Fig. 1. A Reference Architecture for Pervasive Service Ecosystems

At the lowest level is the physical ground on which the ecosystem will be deployed, i.e., a very dense and widely populated infrastructure (ideally, a world-wide pervasive continuum) of networked computing devices (e.g., PDAs, tags) and information sources (Web fragments). At the highest level, service developers, producers and consumers of services and data, access the open service framework for using/consuming data or services, as well as for producing/deploying in the framework new services and new data components. At both levels, the architecture exhibits a high-degree of openness and dynamics, as new devices, users, services, data components can join and leave the system at any time. Between these levels, there are the components of the pervasive ecosystem architecture.

The "Species" level is the one in which physical and virtual devices of the pervasive system, digital and network resources of any kind, persistent and temporary knowledge/data, contextual information, events and information requests, and of course software service components, are all abstracted as "living individuals" (or agents) of the system. Although such individuals are expected to be modeled (and computationally rendered) in a uniform way, they will have specific characteristics very different from each other, i.e., they will be of different "species". The dense population of devices and actors involved at the highest and lowest levels, together with their dynamics, reflect in the presence of a very massive and dynamically varying number of individuals and species.

The "Space" level provides the spatial fabric for supporting individuals, their spatial activities and interactions, as well as their life-cycle. From a conceptual viewpoint, the "Space" level gives shape to and defines the structure of the virtual world in which individual lives. What the actual structure and shape could be, might depend on the specific abstractions adopted for the modeling of the ecosystem. From a practical viewpoint, the spatial structure of the ecosystem will be implemented by means of some minimal middleware substrate supporting the execution and life cycle of individuals, and will enforce concepts of locality, local interactions, and mobility, coherently to a specific structure of the space.

The way in which individuals live and interact (which may include how they produce and diffuse information, how they move in the environment, how they self-compose and/or self-aggregate with each others, aggregate, how they can spawn new individuals, and how they decay or die) is determined by the set of fundamental "Laws" regulating the eternal service ecosystems model. Such laws, or "eco-laws", are expected to act on the basis of spatial locality principles, as in real laws of nature (which is also what makes real ecosystems scalable): the enactment of the laws on individuals will typically affect and be affected by the local space around them and by the other individuals on.

The dynamics of the ecosystem will be overall determined by having individuals in the ecosystem act based on their own internal goals, yet being subject to the eco-laws for their actions and interactions. The fact that the way eco-laws apply may be affected by the presence and state of other individuals, provides for closing the feedback look which is a necessary characteristic to enable self-* features. As far as adaptation over time and long-term evolution are concerned, the very existence of the eco-laws can make the overall ecosystem sort of eternal, and capable of tolerating dramatic changes in the structure and behavior of the species. Simply said in ecological terms: while the basic laws of life (i.e., the basic infrastructure and its laws) are eternal and do not change (i.e., do not require re-engineering), the forms under which it manifests continuously evolve (i.e., the actual service and data species), naturally inducing new dynamics for the interactions between individuals and for the ecosystem as a whole.

2.1 Metaphors and Related Work

The key difference in the possible approaches that can be undertaken towards the realization of eco-inspired service frameworks stands in the metaphor adopted to model the ecosystem, its individuals, the space in which they live, and its laws.

Physical metaphors consider that the species of the ecosystem are sort of computational particles, living in a metric space of other particles and virtual computational fields, which act as the basic interaction means. In fact, all activities of particles are driven by laws that determine how fields should diffuse and how particles should be influenced by the local gradients and shape of some computational field. Physical metaphors have been proposed to deal with several specific middleware-level aspects in dynamic network scenarios [2, 13] or in the area of amorphous computing [16]. However, they appear not suitable for general adoption in large-scale pervasive service ecosystems, due to the fact that they hardly tolerate diversity and evolution (i.e., the number of behaviors and interactions enforced via fields is limited).

Biological metaphors typically focus on biological systems at the small scale, i.e., at the scale of individual organisms (e.g., cells and their interactions) or of colonies of simple organisms (e.g. ant colonies). The species are therefore either simple cells or very simple (unintelligent) animals, that act on the basis of very simple reactive behaviors and that are influenced in their activities by the strength of specific chemical signals (i.e., pheromones) in their surroundings. Biological metaphors are very similar to physical ones. Indeed, they too have been proposed to solve specific algorithmic problems in distributed network scenarios [15], and they too can hardly be suitable for general service ecosystems due to the same shortcomings of physical systems.

Chemical metaphors consider that the species of the ecosystem are sorts of computational atoms/molecules, living in localized solutions, and with properties described by some sort of semantic descriptions which are the computational counterpart of the description of the bonding properties of physical atoms and molecules. The laws that drive the overall behavior of the ecosystem are sort of chemical laws that dictate how chemical reactions and bonding between components take place to realize self-organizing patterns and aggregations of components. Chemical metaphors have been proposed to facilitate dynamic service composition [15], and they appear to well tolerate diversity and evolution. However, they cannot easily apply in large-scale and spatially distributed systems.

In the end, only ecological metaphors which focus on biological systems at the level of animal species and their interactions (as described in the following and along the lines envisioned in [1]), promises to be suitable for large-scale pervasive service ecosystems. In fact, other than supporting adaptive spatial forms of self-organization based on local food-web interactions, they also inherently support diversity and evolution. Others possible approaches that can be suited to model pervasive computing scenarios and that are somewhat similar to the ecological metaphors are market based models [4], these are based on the classical model of consumers, producers and purchasing goods and services. The problem with these approaches is that they can be fitted to pervasive computing, but still there are much work to do to prove this.

3 The Ecological Approach

3.1 Key components

Ecological metaphors focus on biological systems at the level of animal species and of their interactions. In our specific approach, the components of the ecosystem are sort of goal-oriented animals (i.e., agents) belonging to a specific species (i.e., agent classes), that are in search of "food" resources to survive and prosper (e.g., specific resources or other components matching specific criteria) that is, individuals have the ego-centric goal of surviving by finding the appropriate food and resources. The laws of the ecosystem determine how the resulting "food web" should be realized and ruled, that is, they determine how and in which conditions animals are allowed to search food, eat, and possibly produce and reproduce, thus influencing and ruling the overall dynamics of the ecosystem and the interaction among individuals of different species.

The shape of the space is typically organized around a set of localities, i.e., of ecological niches (think at a set of local pervasive computing environments), yet enabling interactions and diffusion of species across niches. Each locality will determine how the different species organize to live, will describe how individuals of each species respond to the distribution of resources and other species, and how they alter these factors.

Such Ecological metaphor promises to be very suitable for local forms of spatial self-organization (think at equilibria in ecological niches), and are particularly suited for modeling and tolerating evolution over time (think at how biodiversity has increased over the course of evolution, without ever mining the health existence of life in each and every place on earth). However, understanding how to properly control the local and global equilibria of real ecological system is a difficult task, and it would probably be very difficult also in their computational counterparts, yet we think that is an interesting approach to explore.

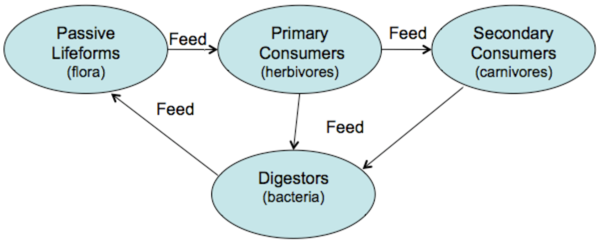

Fig. 2. Key Elements for an Ecological System.

In general, an ecological system can consider the presence of different classes of living forms (see Figure 2). Passive life forms (i.e., the flora system) do not actively look for food, although their existence and survival must be supported by nutrients that are in the space. Primary consumers (i.e., herbivores) need to eat vegetables to survive and prosper. Secondary consumers (i.e. carnivores) typically need to eat other animals to survive, though this does not exclude that can also act as primary consumers (eating vegetables too). The result of the metabolization of food by both primary and secondary consumers ends up in feeding lower-level "digesters" life forms (e.g., bacteria), densely spread in space, and that in their turn produce and diffuse resources and nutrients for the flora.

Let us now translate the above concepts in computational terms. Passive life forms represent the data sources of the ecosystem, which are not to be considered proactive computational entities. Primary consumers represent those services that require to digest information to be of any use, and yet are computationally autonomous. Secondary consumers, instead, are those services that, to be of any use, need the support of other services, other than possibly of information sources. Digesters can be generally assimilated to all those background computational services that are devote to monitor the overall activities of the system, and either produce new information about or influence the existing information. This closes the food-web loop that can support self-organization.

3.2 Modeling of individuals

Let us describe the elements that define an individual of the eco-system. These include: its needs, a happiness function and a set of actions it can perform.

Each individual needs food and resources to survive. These needs are specific for each individual/species. These typically will be represented by a sort of "templates" (possibly a semantic representation) describing the characteristics of the needed resources or individuals. Of course, this also requires that each individual exposes its own characteristics.

Based on the need of an individual, and of the characteristics of the individuals around it, a process of matching takes place. The general template for the match function is given by: $match(individual, individual\_need)$. This function should return some value, expressing the degree of match if in a given niche there is an *individual* and an available *individual_need*.

Based on this match, individuals can, according to specific behaviors, start interacting with each other (e.g. a primary consumer having found matching information, can decide to absorb or consume it). The happiness function tells us the satisfaction degree of each individual, in other words how well an individual satisfies himself. In general, the happiness of an individual is greater when it finds a lot of resources to eat (i.e. lot of matching individuals). Also, the happiness of an individual it matches with, can transitively influence an individual.

The individual's happiness is influenced by the different situations caused by other individuals that are in his near environment. The near environment has a limited area that is regulated by the specific characteristics of the environment. For instance, in a wireless scenario, the near environment can be defined as the space covered by the radio range of an individual.

We can measure two kinds of individuals' happiness during the time: one that is instantaneous and does not consider what happened before and the other is the overall happiness, that considers the instantaneous happiness and the happiness happened in the past.

The instantaneous happiness of an individual is related to his needs in a given moment, which means that it is directly proportional to the match function and the number of individuals that are in his near environment. The general function to obtain the instantaneous happiness of an individual in a niche, where there are n individuals that influence in his happiness is given by:

$$H_{inst}(t) = f(match(individual_i, individual\_need_i), i = 1, ..., n).$$

The general template for the overall happiness function is given by:

$$H_{overall}(t) = f(H_{inst}(t) + H_{overall}(t-1)).$$

The actions are the third element that defines an individual. The actions that an individual perform during the time are always looking to increase his happiness. These actions can be move, reproduce, migrate, dye, etc.

The degree of happiness influences in the actions that each individual performs. Each individual moves through the space until he finds an appropriate niche where he feels well, gets a proper food and reaches some kind of happiness. In other words, if an individual is not happy or wants to be happier (find a better food), then he will move (perform some actions) from one niche to another until finds the appropriate one.

Here we have presented a simple model based on the individuals' happiness in order to choose the appropriate behavior at each tick. There are others similar models [6, 11, 14] based on agents' satisfaction or motivation, but they have not been applied to pervasive computing yet.

3.3 A Case Study

To better clarify this idea, let us present a simple case study we are currently in the process of developing and testing, which has been inspired by [7, 8].

Consider a scenario with diverse kind of devices on it, like a thematic park or an exhibition center, densely pervaded with digital screens where to display information, movies, advertisements, or whatever. We can consider each of these screens (i.e., the computational resources associated with each of them) as a spatially confined ecological niche. Different classes of visitors will watch these screens to look for different information (intended as passive life forms). Thus, we can think at sort of "user agents" executing on the users' PDAs that, once in the proximity of a screen (i.e., while finding themselves into that specific ecological niche) start looking for specific information to eat (i.e., to have it displayed). User agents would thus act as primary consumers with the goal to find the information required by users. Concurrently, we can think at "advertising agents" that, acting on behalf of some advertising company, roam from screen to screen in search of specific classes of user agents (i.e. those interested in specific types of information), with the ultimate goal of displaying advertisements where they could be more effective. Advertising agents would this act as secondary consumers. Background monitoring agents, executing on each ecological niche and possibly interacting with each other, can contribute replicating and spreading information there where it appears to be more appreciated, and can also contribute in supporting the spatial roaming of advertiser agents by directing them there where they could find more satisfaction. Thus, they would act as digesters.

The resulting food web can be summarized as follow (see Figure 3). Information agents feed the user agents, and these feed the advertise agents. Display agents (monitoring agents) maintain the feedback in the system (between information agents and user agents; and user agents and advertise agents), in order to maintain the equilibrium in the ecosystem. It is expected that the whole system continuously move during the time following this food web. In particular, user agents will keep moving trying to follow users movements.

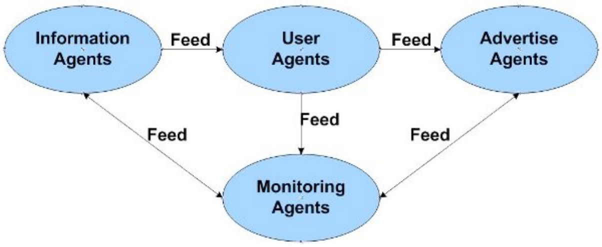

Fig. 3. Food web for an Ecosystem of Displays.

The feedback loop that derives from the above activities can contribute to properly rule the overall dynamics of the screen ecosystem, by continuously self-organizing and self-adapting the way information flows in the system, as well as the way advertising agents move, act, and coordinate with each other. The possibility of exerting control over the dynamics of the system is ensured by the possibility of injecting in the system additional classes of "digester agents" that can radically influence the dynamics of information diffusion and the activities of advertising agents. The adaptation of the system over time is ensured by the fact that it is mostly irrelevant, for the overall functioning of the system, what specific classes of information user agents want, or what the specific goal of advertising agent is. In fact, independently of the specific species of life forms that will populate the system, the basic eco-laws will ensure that such life forms will either find their way of living and their role in the system (e.g., as it can be the case of useful information and of advertising agents that find appropriate users to which to display their ads), or will simply disappear (as it can be the case of useless information or of advertisements no user is interested in).

Let us now detail the modeling of the various agent classes, i.e. of this is the specific happiness evaluation function.

The users' happiness is directly proportional to the match of their information requests, if they were satisfied or not. Then, the formula that gives us the happiness of each user is given by the followings:

$$H_{user}(t_0) = 0. \tag{1}$$

The instantaneous happiness for one user is the following:

$$H_{user}(t) = \sum_{k=0}^{n} match(user, displayedTask_k(t))/n. \tag{2}$$

In this formula, we define n as the quantity of displays that are around the user in a specific time, which are going to influence in his happiness. The $match$ function returns the value of the user's happiness when he sees something in a given display (k) and in a given time (t). This function returns different values depending on what the $displayedTask$ is.

The overall happiness for one user agent is:

$$H_{user}(t+1) = \sum_{k=0}^{t-1} H_{user}(k) + H_{user}(t) - \triangle. \quad (3)$$

The equation has the sum of user's happiness in the past (before t) and the instantaneous happiness. We define the symbol $\triangle$ as a constant value that represents the user's unhappiness, makes his happiness decrease during the time. This value is useful when the user does not see something that he is interested in, in this case, the current happiness will return 0 and this constant will make the happiness decrease.

Advertise agents are happy when they show their publicity in an environment with happy users that want to watch their advertisements. The formula that evaluates the happiness of advertisements is given by:

$$H_{adv}(t_0) = 0. \quad (4)$$

The instantaneous happiness for one advertisement agent is:

$$H_{adv}(t) = (\sum_{k=0}^{n} match(adv, dT_k(t))/n) * (\sum_{k=0}^{m} H_{user}(t)/m). \quad (5)$$

In this function n and m are defined as the quantity of displays and users respectively that are around the advertisement in a specific moment t and influence in the advertisement's happiness. $dT_k(t)$ refers to the $displayedTask$, that is the currently task showed in the displays. The function $H_{user}(t)$ can evaluates the overall happiness average or the instantaneous happiness of the user. If we evaluate the overall happiness then, the advertise happiness will be influenced by all the happiness that had happened before (for example if the user was happy or not in the past). If we evaluate the instantaneous happiness of the user then, no matter what happened before, just matter the current time that we are evaluating. Both functions have their advantages and disadvantages, these will be analyzed with the results of the simulation of each function.

The *match* function returns the value of the happiness of the advertise agent when he sees something in a given display (k) and a given time (t). For example, if the advertise agent sees information in the display, this function should return a positive value, making the advertisement's happiness increase a little, because he is interested in happy users.

The overall happiness for one advertisement is given by:

$$H_{adv}(t+1) = \sum_{k=0}^{t-1} H_{adv}(k) + H_{adv}(t) - \triangle. \quad (6)$$

As in the users' happiness, this function includes the happiness in the past and the current one. $\triangle$ is the value that decreases the advertisement happiness.

The display agents just want to show advertisements because they pay for the service. The happiness of the display is directly influenced by the amount of advertisements showed and by the happiness of their clients (advertise agents). The formula that evaluates the happiness of the displays is given by:

$$H_{dis}(t_0) = 0. \tag{7}$$

The instantaneous happiness for one display agent is:

$$H_{dis}(t) = (\sum_{k=0}^{n} H_{adv}(k)/n). \tag{8}$$

We define n as the number of advertises that are around the display. Here, we have the same problem that we described before. We have two kind of advertise agent happiness that can be analyzed here, the overall and the instantaneous happiness. We plan to analyze it with the results of the simulations.

The overall happiness for one display is given by:

$$H_{dis}(t+1) = \sum_{k=0}^{t-1} H_{dis}(k) + H_{dis}(t) - \triangle. \tag{9}$$

This function is defined as the overall happiness described before.

A possible reason that it causes the decrease of the display agents happiness is that in their niches, there are not enough advertise agents that want to show their publicity, because there are not enough happy user agents because there are not enough information agents. To overcome this situation and reach the happiness, the display agent has to generate more information agents, in order to have happy users and indirectly happy advertisements in his niche.

Displays agents have two main functions: to select the task (information or advertisement) to display and to provide the necessary information to the user and advertise agents, in order to they can move towards the niches where they can find what they need (information and user agents corresponding).

The selection of the task to show is made in each niche by each display agent. This selection tries to maximize the displays' happiness and consequently, the happiness of the advertisements that are around the display (see Formulas 8 and 9). In simple words, the selection consists of choosing the task that will make the advertisements happier.

Display agents examine their niche to provide the necessary information about user and information agents. Before the user and advertise agents decide where to move, they evaluate which is the best next position to find what they want. The evaluation can be based on several considerations:

How? Advertise and user agents require to the near display agents the necessary information to decide in which direction (toward witch niche) move. If they do not find the appropriate users/information around, they move in randomly direction. Where? Just move to the position that is next to their current position. The migration is not considered. When? Advertisements and users remain in their current position unless their happiness is equal to zero ($H = 0$). If their happiness is zero, they evaluate the niches around.

4 Experimental Results

4.1 The simulation environment

We have implemented a simulation environment to assess our approach in the context of the case study, within the Recursive Porous Agent Simulation Toolkit (Repast).

As from the Section 3.3, the scenario considers a 2D space (the thematic park), where different spatially situated agents (users, advertisements, information and display) act, interact and evolve given the eco-laws (trying to maximize their happiness). Simulations initialize user, advertise and information agents with randomly positions. User and advertise agents can move during the simulation to find what they want: information and user agents respectively. Displays agents have a fixed position during the whole simulation. We particularly study the behavior of users, advertise and displays.

The simulation environment has currently some limitations that will be overcome in the future, and that nevertheless enable us some preliminary assessment of our approach. The current limitations are: (*i*) User agents do not move continuously but stop when they find their appropriate niche. (*ii*) Agents do not contemplate the dynamical actions of joining or leaving the system and the overall number of agents is static. (*iii*) Information and advertise agents do not self adapt to the users requirements. (*iv*) Agents cannot migrate, reproduce or dye.(*v*) Display agents do not interact with each other.

4.2 Results

Several experiments have been performed to test the behavior of the eco-system of displays in our simulation environment.

Figures 4 and 5 show two screenshots of a simulation scenario with 100 advertise agents, 100 user agents, 100 information agents and 25 displays. The squares represent the displays and the little bodies represent the users. Displays can show different colors that represent what they are showing. In order to make clearer these Figures, we do not show advertise and information agents.

These Figures show how the user agents' behavior changes during the time, particularly how they group around displays. The Figures were taken at two different moments of the simulation: the first one at the beginning, when users move through the space (Figure 4) and the second one almost at the end, when users find the niche where they feel happy (Figure 5). After a period of time, we can see that agents always get grouped in the niches where they find what they need, i.e. they feel "happy". Agents move, flow through the space until they find their appropriate niche.

To evaluate the whole behavior of the system, we measured the happiness of agents and their process of convergence. Figures 6 and 7 show how the current average happiness (between 0 and 1) of users and advertisements varies over time until it reaches the equilibrium. The happiness of display agents is not showed, because their happiness is proportional and quite similar to the advertise

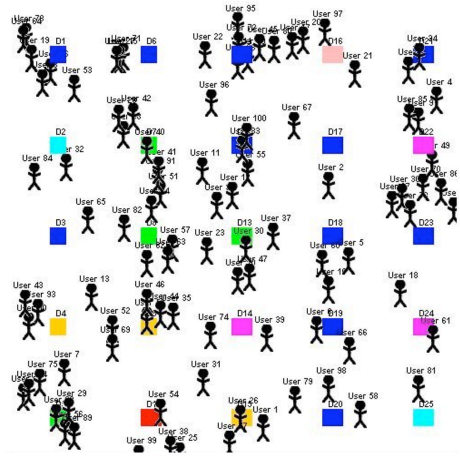

Fig. 4. Space View for 100 agents (few ticks).

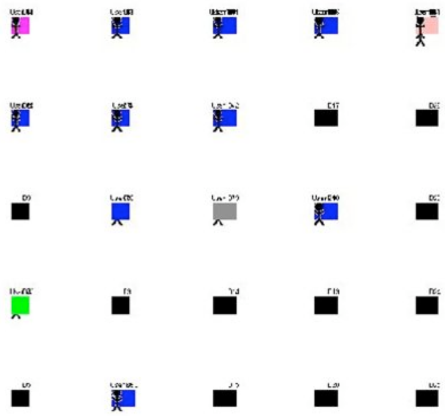

Fig. 5. Space View for 100 agents (the equilibrium).

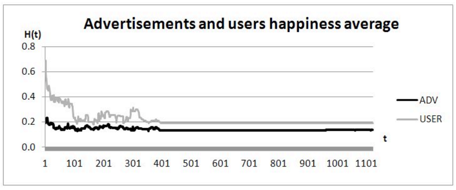

Fig. 6. Happiness average for 100 agents.

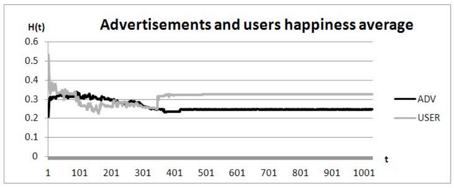

Fig. 7. Happiness average for 500 agents.

agents happiness. We can see that the curve fluctuates at the beginning, as user and advertise agents are looking what they want (information and users respectively), when they begin to find their appropriate niches, the curve begins to stabilize, until arrives to be almost a constant (not change during the time), that means that the systems arrives to a balance. These Figures show clearly how the agents can organize by themselves and drive the system to a balance state, independently of the actual size of the system. The simulated eco-system has a great potential to deal with the global self-organization of the activities of their individuals (agents).

By aggregating the results of several experiments with varying number of agent, we had similar results and conclude that the system always reaches the equilibrium, no matter the number of components. In other words, we can say that system promises to be scalable. This is showed in Figures 8 and 9. Figure 8 shows how many ticks approximately the systems need to reach the equilibrium under different number of agents. Advertisements and users almost always arrive at the same time to equilibrium. Figure 9 shows the happiness values of users and advertisements when the system reaches the equilibrium state. It seems like the happiness of both agents grows if the quantity of user agents grow, we need to do more test in order to confirm this trend. This would be a good point for the scalability of the systems: the more user agents are, better the overall organization of the system.

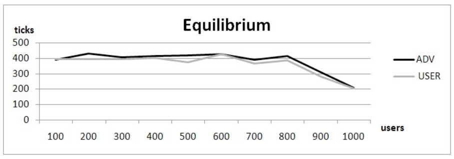

Fig. 8. Quantity of user by quantity of ticks to arrive the equilibrium.

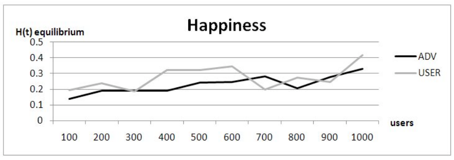

Fig. 9. Quantity of user happiness in the equilibrium tick.

5 Conclusions

In this paper, we have elaborated on the idea of model and develop next generation of pervasive service framework inspired in the ecological model. That is, of conceiving future pervasive service frameworks as a spatial ecosystem in which services, data items, and resources are all modeled as autonomous individuals (agents) that locally act and interact in accord to a simple set of well-defined “eco-laws”.

We attempted to clarify these ideas through a simulation of a case study related to an adaptive advertisement displays. The development of this simulation is not finished yet, however the first results give some feedbacks about the self-organization, the scalability and process of convergence of the system.

Nevertheless, there are still a lot of experiments to do and several open questions to answer to better assess our approach: (*i*) Concerning to the simulation environment: we have to extend it to overcome its current limitations. (*ii*) Concerning to the general applicability of approach: we have to experience with a larger variety of case studies. (*iii*) Finally, we have to better formalize and generalize the model and realize a prototype implementation.

References

1. Gul Agha. Computing in pervasive cyberspace. *Commun. ACM*, 51(1):68–70, 2008.
2. Jacob Beal and Jonathan Bachrach. Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intelligent Systems*, 21(2):10–19, 2006.
3. Gabriella Castelli, Alberto Rosi, Marco Mamei, and Franco Zambonelli. A simple model and infrastructure for context-aware browsing of the world. In *PERCOM '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pages 229–238, Washington, DC, USA, 2007. IEEE Computer Society.

4. David Cornforth, Michael Kirley, and Terry Bossomaier. Agent heterogeneity and coalition formation: Investigating market-based cooperative problem solving. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 556–563, Washington, DC, USA, 2004. IEEE Computer Society.
5. Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, 2002.
6. Daniel Thalmann Etienne de Sevin. An affective model of action selection for virtual humans. In *AISB '05: Proceedings of Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action symposium at the Artificial Intelligence and Social Behaviors*, 2005.
7. Alois Ferscha, Andreas Riener, Manfred Hechinger, and Heinrich Schmitzberger. Building pervasive display landscapes with stick-on interfaces. In *Workshop "Information Visualization and Interaction Techniques", associated with CHI'06 International Conference, Quebec, Canada*, page 9, April 2006.
8. Alois Ferscha and Simon Vogl. The webwall. In *Proceedings of the Ubicomp 2002 Workshop on Collaboration with Interactive Walls and Tables*, Göteborg, Sweden, September 2002.
9. Sebastian Herold, Holger Klus, Dirk Niebuhr, and Andreas Rausch. Engineering of it ecosystems: design of ultra-large-scale software-intensive systems. In *ULSSIS '08: Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems*, pages 49–52, New York, NY, USA, 2008. ACM.
10. Michael N. Huhns and Munindar P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, 2005.
11. Pattie Maes. The agent network architecture (ana). *SIGART Bull.*, 2(4):115–120, 1991.
12. Marco Mamei, Ronaldo Menezes, Robert Tolksdorf, and Franco Zambonelli. Case studies for self-organization in computer science. *Journal of Systems Architecture*, 52(8):443–460, 2006.
13. Marco Mamei and Franco Zambonelli. *Field-Based Coordination for Pervasive Multiagent Systems (Springer Series on Agent Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
14. Jacques Ferber Oliver Simonin. Modeling self satisfaction and altruism to handle action selection and reactive cooperation. In *SAB '00: Simulation of Adaptive Behaviors*, pages 314–323, 2000.
15. Raffaele Quitadamo, Franco Zambonelli, and Giacomo Cabri. The service ecosystem: Dynamic self-aggregation of pervasive communication services. In *SEPCASE '07: Proceedings of the 1st International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments*, page 1, Washington, DC, USA, 2007. IEEE Computer Society.
16. David Servat and Alexis Drogoul. Combining amorphous computing and reactive agent-based systems: a paradigm for pervasive intelligence? In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 441–448, New York, NY, USA, 2002. ACM.
17. Mihaela Ulieru and Steve Grobbelaar. Engineering industrial ecosystems in a networked world. In *5th IEEE International Conference on Industrial Informatics*, pages 1–7. IEEE Press, 2007.
18. Roy Want. An introduction to rfid technology. *IEEE Pervasive Computing*, 5(1):25–33, 2006.