Social Intelligence as the Means for Achieving Emergent Interactive Behaviour in Ubiquitous Computing Environments

Ioannis D. Zaharakis[1], Achilles D. Kameas[1,2]

[1]Research Academic Computer Technology Institute
N. Kazantzaki str, University Campus,
26500 Patras, Hellas
{jzaharak, kameas}@cti.gr

[2]Hellenic Open University
23 Sahtouri str,
26222 Patras, Hellas

Abstract. This work introduces a framework for modelling the main actors (human, artefacts and services) in a symbiotic Ambient Intelligence environment. It, also, proposes an architectural scheme that associates the social behaviour, which is not an inherent characteristic of the participants, during interaction with the functional behaviour of the participants of a Ubiquitous Computing application. The overall approach is demonstrated by a specific example of application which illustrates its concepts through a more technical point of view.

Keywords: Ambient Intelligence, Emergent Behaviour, Human-Computer Interaction, Social Intelligence, Ubiquitous Environments.

1 Introduction

As computational power diffuses in our living/working environment and the everyday devices that are capable of sensing, processing and communicating continuously grow in number, the potential use of the objects emerges mainly from the interactions of the humans with the digital devices and these interactions are not only time-dependent but also space- or context-dependent. A further consequence is that the nature of the human activities eventually assisted by artefacts is rapidly changing. Execution of tasks involving the use of (collections of) artefacts may become difficult due to the inherent systemic complexity of Ubiquitous Computing (UbiComp) applications, which, among others, results from device incompatibility, and the huge number of interactions among visible and non-visible actors.

There is a pressing need for a design framework that will act as a common referent between designers and users (i.e. ordinary people) of UbiComp applications that exist within an Ambient Intelligence (AmI) environment. This framework should support not only the representation of user's interactions with artefacts or predefined collections of artefacts (such frameworks already exist today, see [8]), but would also cater for the design of adaptive interactions, as artefact ecologies will be evolving to encompass changing user requirements. Finally, the supported interactions should be people-centred, because with an AmI environment (according to ISTAG vision of AmI [7]) people will be acting as naturally as possible (while now they are inter-acting with a computer).

In this paper, we provide a new perspective by enabling socially intelligent interactions among people and artefacts and we claim that by subsuming task optimization to social intelligence, people interactions with artefacts will become more natural. This work adopts and extends the framework proposed in [15] which deals with different perspectives of the interrelations developed in symbiotic ecologies where people and artefacts coexist. We propose a subsumption architecture that supports the integration of social behaviour with functional behaviour of UbiComp applications. Our approach is bottom-up, in the sense that it considers social and functional elementary "behaviours" as basic building blocks of complex and emerging UbiComp behaviour.

The main innovation introduced by the proposed framework is that we do not aim simply at autonomous systems; instead, we aim at systems that although consisting of largely reactive parts, they exhibit a pro-active behaviour in a social level. Furthermore, the social behaviour is not an inherent characteristic of the participants but it is a result of the interactions among participants.

The next section details on the proposed framework integrating people, artefacts and services into a symbiotic AmI environment. The section 3 focuses on interaction with UbiComp applications and on how the social and the individual behaviours are reflected by the participating artefacts and the provided services. Finally, the section 4 puts our approach into an example scenario and demonstrates a realisation of the proposed concepts through a more technical portion.

2 Proposed Framework

Technically, the reproduction of social behaviours and the handling of complex tasks with an equal agility as the one exhibited by natural intelligent systems could be achieved by i) considering that all the necessary information lies out the environment and surrounds the participants (according to R. Brooks [2, 3, 4]) and ii) using bio-inspired approaches in designing intelligent systems, in which autonomy, emergence, and distributed functioning are promoted [1, 10].

We propose to distribute the individual physical/computational/cognitive capabilities over the entire ecology and then immerse the ecology into a UbiComp environment, aiming to generate theory and technology for the understanding of the own self and its relation with the surrounding world. We deal with this consideration by i) attributing AmI objects with physical expression (dimensions, shape, texture, colour, plugs, sockets, connectors, etc) and ii) dealing with the provided services as basic behavioural building blocks of the overall system behaviour.

According to our approach, a living/working AmI space is populated of many heterogeneous objects with different capabilities and provided services. All these objects and services are regarded as basic building blocks having an internal part that encapsulates the internal structure and functionality, and an external part that manifests the capabilities and influences the surroundings. Additionally, every basic building block has several pre-defined functions (we call them *basic behaviours*). Some of the basic behaviours are just reactions to external events and some are continuously pursued to be fulfilled. The former type of basic behaviours imitates the reflex actions of the living organisms, while the latter the preservation instincts. The interrelationships between the basic building blocks and the associated environment form an *ecology*. In such ecologies artificial entities coexist unobtrusively with humans and perform collaborative tasks through a continuous evolvable process concerning both their physical and social cognitive growth (we call them "*ambient spheres*"). Similar approaches can be found in a natural ecosystem and have been used in (purely reactive) collective robotics. In addition, we integrate them into another ingredient, called *ambient system* (AmS).

The AmS acts as the "glue" between the tangible and non-tangible basic building blocks of the ecology by providing an Interface Definition Language (IDL) and thus integrating the basic building blocks into a common interoperability framework. One step to this end is the design and implementation of a hierarchy of multi-dimensional ontologies that include both non-functional descriptions, and rules and constraints of application, as well as aspects of dynamic behaviour and interactions. During the ecology life-time a core ontology, open and universally available and accessible, is supplemented with higher-level goal, application and context specific ontologies. These ontologies describing specific application domains can be proprietary. Emerging behaviour, in this context, are considered as a result of interactions among heterogeneous, seemingly incompatible or non pre-defined entities. Additionally, the AmS is responsible for the observation and collection of the interactions between AmI objects, provided services and users. The collected data are used to create best practice ecology configurations that will help the gradual assembly of social memory. Aspects of Social Intelligence are embodied to the ecology configurations as basic social behaviours aiming to regulate the group interactions. These social behaviours are provided as ontological constructions which are also subject to evolution. On the other hand, all the collected information are provided by the AmS as feedback to the members of the ecology (social memory in association to user "perceived quality") favouring the best configurations and implicitly assigning cognition to the whole ecology. It is mentioned that the feedback information is provided to the ecology as another stimulus or stimuli and thus does not require extra (complicated) sensors mounted to the artefacts. Notably, this would require a different modelling and engineering approach comparatively to the one described so far. Instead and along the lines of Swarm Intelligence where the environment is a stimulus for the swarm, we treat the AmS as another (special) basic building block, the environment building block.

The AmS system is realised as a distributed platform that supports the instantiation of ambient spheres, each of which is formed to support human activity. Examples of candidate AmS technologies are the currently available distributed component frameworks and service-oriented architectures. An ambient sphere is an integrated autonomous system realised within AmS as a set of configurations between the AmI objects and their services. In our model of discourse, the end-users are placed inside the ambient sphere, as this allows us to model them as another basic building block that generates events and changes the environment.

Focusing on interaction within such an AmI space, people still have to realize their tasks, ranging from mundane everyday tasks (i.e. studying, cooking etc) to leisure or work related tasks, or even tasks that relate to emergency situations (i.e. home care, accident, unexpected guests etc). To do so, they have at their disposal the objects that surround them. These in fact, are new or improved versions of existing objects, which by using Information and Communication Technology (ICT) components (i.e. sensors, actuators, processor, memory, wireless communication modules) can receive, store, process and transmit information, thus allowing people to carry out new tasks or old tasks in new and better ways. In the following, we shall use the term "artefacts" for this type of augmented objects. Turning an object into an artefact is a process that aims at enhancing its characteristics and properties and abilities so that the new affordances will emerge. In practical terms, it is about embedding in the object the necessary hardware and software modules.

From the interaction point of view, we are mostly concerned with the interface of the artefacts and the collective interface of UbiComp applications. The former shall directly affect or depend upon the physical form and shape of the artefacts. The latter can exist in the "digital space" of a computer, i.e. a PDA that runs specific software representations of the artefact services. Thus, people interact with an AmI environment in order to [9]:

- Engineer a UbiComp application within the environment, as a composition of artefacts, which collectively serve a specific purpose or satisfy a declared set of needs.
- Use an application to satisfy their needs: such an application may be composed by people themselves, or could be bought and installed.

Interaction takes place in two levels:

- Artefact-to-artefact: the objects themselves may form an "underlying" layer of interactions, mainly in order to exchange data and to serve their purpose better. Such interactions may use wired channels or any of the available wireless protocols (in a peer-to-peer or broadcast manner), or even the Internet.
- User-to-application. The user interacts a) with any single artefact b) with a collection of co-operating devices. Moreover one has to consider the case where many users interact with the same application.

The degree of visibility and control that people may have on these interactions may vary depending on people's ability to perceive the system state: any of these two types of interaction may happen either explicitly or implicitly.

An explicit interaction happens under the control of people always provides feedback about its state to them. Although this may seem desirable, it may also become very annoying if one takes into account that there will be hundreds of artefacts in our environment. People interact explicitly with objects or services (or collections of these). In the case of individual objects (or, preferably, services), interaction can be supported by the object affordances. When people interact with a UbiComp application composed of a collection of objects, its "affordances" have to emerge and made explicit.

Implicit interactions are usually under the control of actors other than people; these could be processes, artefacts, intelligent agent mechanisms or even artefact owners. Implicit interactions can only be acceptable if they can be trusted and do not violate privacy or ethics. People need not be directly aware of the communication among objects. Moreover, even certain interactions with UbiComp applications should happen unobtrusively, i.e. people should be made aware of the state changes of the application components without being disrupted from their current tasks.

Finally, one should also consider the context of interaction, which ranges from public to private (with respect to disclosure), from individual to shared (with respect to stakeholders) and from closed to open (with respect to space).

3 Interacting with AmI Spheres

The AmI spheres constitute a dynamic distributed system, composed of artefacts with finite sets of capabilities (services) offered usually through proprietary user interfaces. People have to interact with an AmI sphere in two levels:

- the task level, whereby they will have to use each individual artefact in order to make use of the collective AmI sphere capabilities
- the meta-task level, whereby they will have to compose, decompose or otherwise edit AmI spheres

When interacting with an AmI sphere, people are in fact using the artefacts that compose it (i.e. they are simply acting, not interacting). This is as close as we can get to the notion of calm technology promoted by M. Weiser, who stated that the most profound technologies are those which disappear in the background [14]. This view is directly inspired from Heidegger's theory of "dasein", which states that people are thrown in the world and are always engaged with acting within it to accomplish their tasks. In this view, technological tools disappear in the background in favour of tasks-at-hand; tools only appear when the task accomplishment procedure breaks down, that is, when something goes wrong.

First of all, artefacts have to demonstrate their affordances, both in the physical environment (for people to be able to use them) and to the digital space (so that other artefacts, agents etc will be able to interact with them). Then, the state of each artefact must be made visible / available for the same reasons (although the procedure used to compute the state should be internal to the artefact). AmI spheres introduce two factors that cause the breakdown of the existing task models: people will have to make sure that they can still carry on with ordinary tasks, and they will have to become familiar with the new affordances of the artefacts. In addition to adapting their skills for using artefacts, people will have to develop skills for using the computing properties of their new environments as well [12].

3.1 The GAS Approach

To deal with the above, Gadgetworld Architectural Style (GAS) adopts a layered architecture that transparently supports composing and using AmI spheres (called eGadgetWorlds) from autonomous artefacts (called eGadgets), which can be objects, services or both [8]. To enable composition of AmI spheres, GAS proposes the plug-synapse model: a "plug" is the manifestation of a property, capability or service in a semantically-rich way and a "synapse" is a communication established between compatible plugs. For example, a TV set may offer the display service and a digital camera may establish a synapse in order to output images; a chair may offer the capability to recognize whether a person is seated on it and a table lamp may use it to switch itself on; etc. Plugs are either ingoing (i.e. used as "remote sensors") or outgoing (i.e. used as "remote actuators"). Clearly this concept scales well, as more complex "plugs" can be defined as compositions of simpler "plugs", either at an artefact or sphere level. For example, a "reading plug" for an office AmI sphere may be defined by combining specific plugs from a chair, a table, a lamp and a book (in fact, we refer to their artefact counterparts) in such a way that when someone is seated on the chair and the chair is located close enough to the table and a book is opened on the table, then the lamp is switched on; with the help of the room (considered as an artefact), the system could also recognize who is seated on the chair and switch on automatically his/her reading profile; then, the "reading plug" could be used by the room to redirect phone calls so as not to disturb the user unless necessary.

An interesting case appears when the AmI sphere breaks down. Consider, for example, the case where the desk lamp is broken. Then the system can either inform the user and wait for his/her action, or search for a similar service in the environment (for example, the sphere, with the help of the AmS system, can locate the room lamp and switch it on). In the latter case, it is necessary that all artefacts hold an internal description of their services and goals; and that these descriptions are compatible. GAS includes a multi-layered ontology, which describes artefact "plugs" and rules of usage (i.e. constraints) using a commonly available core ontology of basic terms. The use of ontology makes possible the communication between heterogeneous eGadgets and helps in achieving a shared understanding (as described in [6]). Emergent behaviour of this type is a direct result of the ability of eGadgets to communicate in socially meaningful ways, as described in their hierarchy of basic behaviours.

The above definition supports emerging functionality because (a) artefacts are self-sufficient and their plugs are described in a functionally independent way, (b) not all synapses need to be known from the start, (c) new synapses may be added or existing synapses may be deleted, for example, as an artefact may move outside the sphere (that is, outside the range of the wireless network), and (d) experience may be recorded in local artefact ontologies and appear in the form of higher level plugs (the use of a common core ontology available in the sphere's environment ensures the compatibility of plug definitions).

Plugs and synapses are managed by GAS-OS, a distributed middleware platform that takes care of resource management and communication. Thus, an AmI sphere is defined as a GAS-OS application; all eGadgets in it run GAS-OS; compatible plugs of these eGadgets are engaged in synapses to provide collective sphere functionality.

3.2 Supporting Tasks and Meta-Tasks

Within an AmI sphere composed as an eGadgetWorld, a user may perform his/her tasks simply by using the artefacts therein. We do not propose to embed screen-based interfaces on every artefact, or to use a computer as a sphere master, as these would greatly alter the affordances of the artefacts and consequently have a negative effect on people's capacity to form new task models. Another unwanted consequence is that these artefacts would no longer be functionally autonomous.

In the case of AmI spheres, the issues are to conduct a coherent dialogue composed of user actions within the sphere and eGadgets' responses and to preserve a distributed but meaningful dialogue state. To achieve this, each eGadget must be aware of the state of other eGadgets in the sphere. This can be achieved by exchanging information through the synapses that compose the sphere. By processing the combined perception of the states of itself, peer artefacts (connected via synapses) and the environment and then applying its architecture of subsumed behaviours, each artefact is able to locally maintain a dialogue state that is compatible with the AmI sphere dialogue state.

GAS offers a set of tools that support the meta-tasks of creating and editing spheres. These editors run on PDAs and laptops and have been positively evaluated by non-expert users [13].

3.3 The Subsumption Architecture

The proposed conceptual framework extends the GAS approach, by allowing the eGadgetWorld management tasks to be dynamically performed by the AmS (though the direct user involvement is not prohibited) based not only on the available artefacts and services, but also on the observed user habits and the social rules in context. The proposed approach achieves uniformity and coherence, because it uses the notion of basic behaviours to represent both functional and social capacities of eGadgets, thus providing uniform support for individual functionality and "social interactions" between the sphere members (artefacts, agents, and people).

In order to realise this approach, the decision making module of the middleware of each eGadget is composed of two behaviour calculation modules: those that implement its core functionality and those that realize the context of social intelligence. The former use the data gathered by the eGadget sensors to calculate the object's state and to decide a set of (re)actions. The latter use sensor data as well as synapse input to determine context of operation and to select the most appropriate action in the list.

There are different subsumption architectures to choose from. In the simplest one, all basic behaviours are placed in the same hierarchy. To enhance modularity and scalability, one could separate the basic behaviours into two different subsumption schemes: the social and the functional one. In any case, we choose to give social intelligence behaviours greater precedence over functional behaviours, thus allowing the eGadget to realise the most socially intelligent response depending on the situation at-hand. Thus, using the two level selection mechanism, we ensure independence in the determination of local state and response, while we achieve a socially-driven eGadget behaviour.

Examples of functional basic behaviours are: "turn light on", "produce specific sound", "move towards a specific direction", etc. These depend on the actuators and the outgoing plugs of the eGadget and determine the affordances the eGadget offers in physical and digital spaces. This set also contains the basic behaviours "form synapse" and "learn", which ensure that affordances such as composeability and changeability are supported. To deal with possible heterogeneity in data definition, each eGadget uses a local ontology to translate data incoming through synapses.

Regarding social behaviour, we can choose from the basic social behaviours drawn from the social intelligence studies (e.g., benevolence, non-intrusion, altruism, responsibility, antagonism, empathy, emergency, etc), depending on the nature of the UbiComp application. The subsumption scheme contains in a hierarchical structure several degrees of sociality – from non sociality at all in the lower level to high sociality in the top level – as well as the suppression between the social behaviours.

3.4 The GAS Symbiotic Interaction Metaphor

In order to describe our proposed metaphor for interacting with UbiComp applications composed from communicating artefacts (Fig.1), let us first make explicit some basic assumptions:

- User inhabits an AmI space, which contains artefacts, having a physical presence and offering digital services;
- User forms a plan to achieve a goal he has in mind; in this stage, the plan probably consists of steps and sub-goals, some of which may not be conscious at all;

- User tries to realise his plan by combining services offered by the artefacts in his environment; in this endeavour, user can only be aware of the affordances of the artefacts and tries to use them accordingly

First of all, GAS supports the following meta-tasks, using the GAS editor:

- User can query the services and capabilities of each eGadget
- User composes an eGadgetWorld by combining eGadgets capabilities using the plug/synapse model
- User can have an overview of the existing eGadgetWorlds and even edit or delete any of them

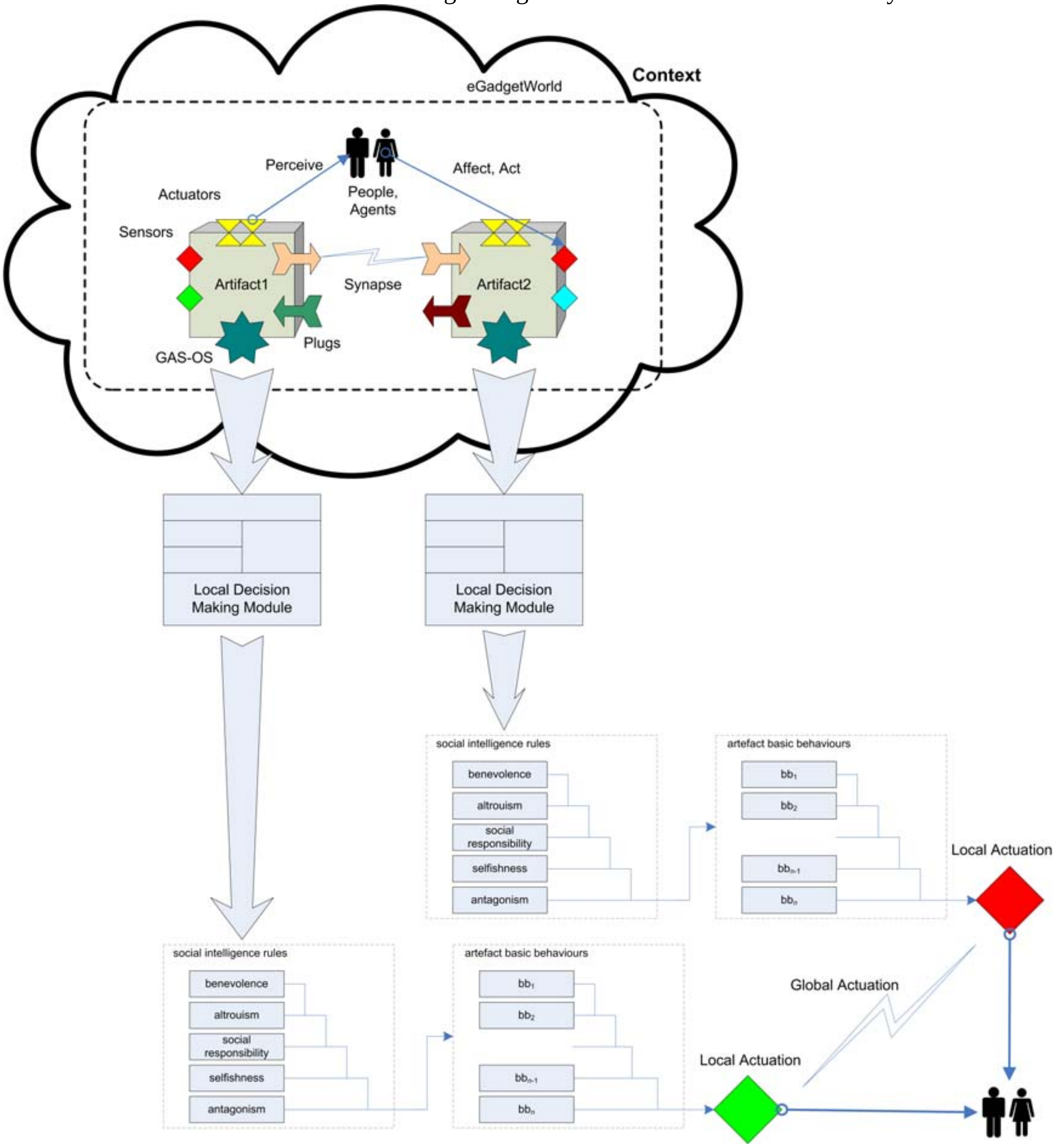

Fig. 1. The proposed architectural scheme supporting the integration of social with functional behaviour of UbiComp applications.

Then, we propose the following interaction metaphor:

- The state of each eGadget is communicated using its actuators
- User simply uses each eGadget based on its affordances, directly affecting its local state
- As a consequence, the eGadget communicates the new state to the eGadgets it is connected to via its synapses
- Peer eGadgets calculate new local states (thus user indirectly affects them) and communicate them using their actuators; each eGadget decides the form of communication using its local subsumption architecture, its local state and the context it perceives via its local sensors and peer eGadgets states (via its synapses)

- A new global eGadgetWorld state emerges as a consequence of local state changes of all the eGadgets in the eGadgetWorld
- The new global state is collectively but independently communicated to the user by all eGadgets in the eGadgetWorld

In this approach, we must make a few remarks. Firstly, because this is a symbiotic ecology, there is no centralized decision making component. All eGadgets are considered as peers and each one is responsible for local decision making and acting by taking into account local and global information (here "global" is restricted to those eGadgets in the AmI sphere). However, the eGadgets that compose an eGadgetWorld, if they have this basic behaviour, may choose to elect representatives, that is, eGadgets which will act as eGadgetWorld "leaders". This process is supported by contemporary communication protocols (i.e. Wi-Fi) and ensures that the eGadgetWorld will remain functional even if some secondary or weak eGadgets are not occasionally operating.

Secondly, because the local eGadget state is communicated to other peer eGadgets via the synapses and triggers changes in their local states, which are also communicated to peer eGadgets, there exists the risk of the eGadgetWorld falling in an infinite loop of recursive global state changes. This falls within the scope of our modelling approach and can be avoided using two measures. Firstly, synapses are directed: if a synapse exists from eGadget A to eGadget B, this means that only changes in the state of eGadget A are communicated to eGadget B and not the other way round. In addition, when composing or editing an eGadgetWorld, the GAS tools offer the user the ability to send a "ping" message, which propagates to all eGadgets in the eGadgetWorld, thus helping in discovering if any loops exist.

4 An Example Scenario

Let's consider a scenario of a not so distant future, whereby we can demonstrate the concepts and potential of the proposed approach.

Anna is a single, 36 years old mother, living with her daughter in a modern 3 room apartment in Athens. She is a hard working employee in a private telecommunications company. Despite the demanding job and family requirements, Anna likes to be a calm person. She likes the view from her apartment and the Greek sunlight. She enjoys a warm morning bath while listening to soft rock music. That is why, before she gets up into her overfull daily schedule, she has programmed her apartment to implement a specific routine. In fact, she has created a "wake up" AmI sphere consisting of the following eGadgets: an alarm clock, the bed-mattress, the bedside picture frame, the bathroom mirror, the window blinds, and the room light. The eGadgets in this sphere have plugs through which they offer access to their properties and services. For example, the clock, the blinds and the room light are equipped with light sensors and can produce a light intensity reading through their "luminosity" plug; the clock also uses the "alarm" plug to send a buzzing event; the bed-mattress has weight and pressure sensors and can decide whether there is someone "lying upon" or not; the window blinds can be set to a specific height or angle through their "lift" and "rotate" plugs.; the bedside picture frame and the bathroom mirror can display images or videos that receive through their "content" plug; finally, the room light also offers the plug "light intensity".

Anna has programmed this eGadgetWorld to gradually increase the intensity of light in her bedroom after the alarm has gone off, to start playing nice relaxing music and display pictures from her summer holiday before it starts communicating to her the first news of the day.

She has managed to program the control of light intensity by connecting the "alarm" plug to the blinds "lift" plug, the light "intensity" plug and the frame "content" plug. With manipulation of synapse rules, the AmI sphere can now, once the alarm has gone off, gradually lift the blinds (until the pre-programmed level of light intensity in the room has been reached), display pictures in the frame and use it to play some music (until Anna has got off from bed, as can be communicated by its "lying upon" plug). Then, as she enters the bathroom, the news of the day are displayed in her bathroom mirror using its "content" plug.

This functionality can be implemented using only the functional basic behaviours of the eGadgets. When the social basic behaviours are also used, then this functionality can become adaptive. For example, because of the "selfishness" basic behaviour, all eGadgets try to achieve their local goals, whereas because of the "social responsibility" basic behaviour, in case the window blinds cannot be lifted due to malfunction, the eGadget will try to rotate the blinds or as a final resort, ask the room light to gradually increase its light intensity, to make up for the blocked sunlight. Additionally, because of the "altruism" basic behaviour, only the blinds can be lifted, if the day is bright, but if the day is cloudy, the room light is also gradually turned on, as it senses a low level of luminosity in the room. Also, the bathroom mirror will take over from the frame the task of communicating the news of the day, when Anna enters the bathroom. On the other hand, because of the “non disturbance” basic behaviour, the blinds are slowly rotated or lifted, the room light gradually becomes brighter and the volume of music gradually increases to the specified levels. At last, when the “emergency” basic behaviour is triggered, a picture of Anna’s partner pops up in the bathroom mirror instead of the news of the day: it’s Valentine’s Day and he recorded a tender message for her!

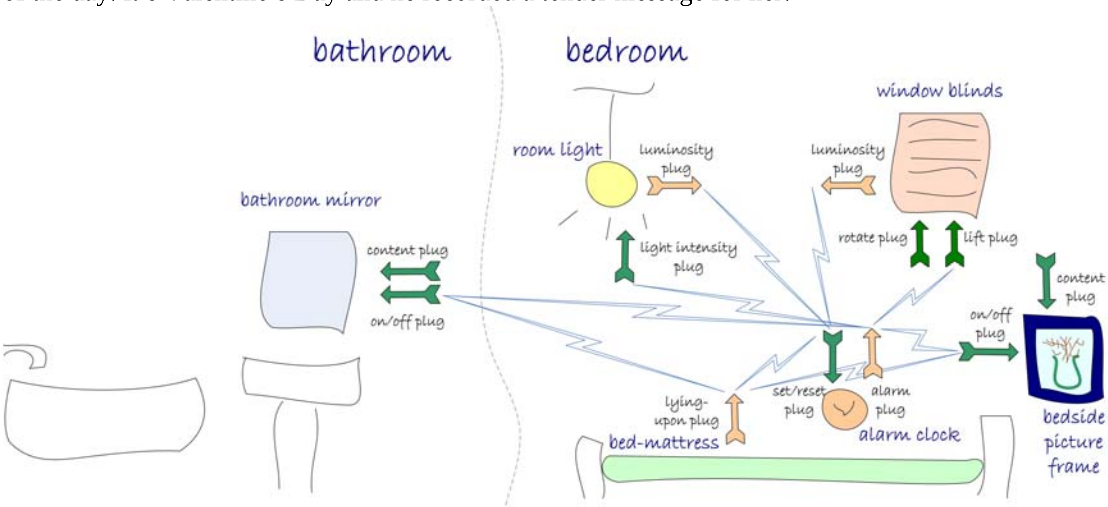

Fig. 2. The “wake-up” AmI sphere.

5 Summary

This work examines symbiotic UbiComp environments and deals with the participating actors and the interactions taking part therein through a novel perspective. Especially, participating artefacts and provided services compose AmI spheres supporting several user tasks taking into account both functional properties and social rules of behaviour. The emphasis is given to social behaviour where social rules suppress individual behaviour when applied. The spheres are dynamic in nature ensuring application efficiency and service provision continuity. However, we investigate methods of providing feedback to the spheres aiming at the creation of a kind of social memory. With the embodiment of machine learning and evolution mechanisms into the ambient system, this phenomenon could be exploited to contribute to the creation of systems with self-management properties such as self-configuration, self-healing, self-optimisation, and self-protection.

Acknowledgement. This work was supported in part by a grant from the European Community under the “Information Society Technologies” Programme (01/05/2006-30/04/2009), Project ASTRA IST-29266 (www.astra-project.net).

References

1. Bonabeau, E., Dorigo M., Theraulaz G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, (1999).
2. Brooks, R. A.: Intelligence without Reason. In Proceedings of 12th Int. Joint Conf. on Artificial Intelligence, Sydney, Australia, (1991), 569 – 595.
3. Brooks, R. A.: Intelligence without representation. Artificial Intelligence, 47:139–159, (1991).
4. Brooks, R. A.: Robot: The future of flesh and machines. London: Penguin Books, (2002).
5. Calvary, G., Coutaz, J., Thevenin, D.: A unifying reference framework for the development of plastic user interfaces. In Proceedings of EHCI01, Toronto, LNCS 2254, (2002), 173-192.
6. Habermas, J.: The theory of communicative action (Vol I and II) (translator McCarthy, T.), Boston: Beacon Press, (1984).
7. IST Advisor Group: Scenarios for Ambient Intelligence in 2010. European Commission Community Research, (2001). Available in from http://cordis.europa.eu/ist/istag-reports.htm

8. Kameas A., Bellis S., Mavrommati I., Delanay K., Colley M., Pounds-Cornish A.: An architecture that treats everyday objects as communicating tangible components, in Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, (PERCOM2003), Texas-Fort Worth, (2003).
9. Kameas, A., Mavrommati I., Markopoulos, P.: Computing in tangible: using artifacts as components of Ambient Intelligent Environments. In Riva, G., Vatalaro, F., Davide, F., Alcaniz, M. (eds.): Ambient Intelligence: The evolution of Technology, Communication and Cognition, IOS Press, (2004), 121-142.
10. Kennedy, J. Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, (2001).
11. Mavrommati, I., Kameas, A.: The evolution of objects into hyper-objects. Personal and Ubiquitous Computing 779, ACM, 7 (3-4): 176-181, (2003).
12. Mavrommati, I., Kameas, A.: The concepts of an end user enabling architecture for ubiquitous computing. In Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive 2004), Vienna, (2004).
13. Mavrommati I., Kameas A., Markopoulos P.: An Editing tool that manages device associations in an in-home environment. In Proceedings of the 2nd International Conference on Appliance Design (2AD), HP Labs, Bristol, UK, 11-13 May 2004, 104-111.
14. Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. CACM, 36(7), July 1993, 75-84.
15. Zaharakis I. D., Kameas A. D.: Emergent Phenomena in AmI Spaces. In Proceedings of 1st International EASST-EU Workshop on Future Research Challenges for Software and Services, associated to The European Joint Conferences on Theory and Practice of Software (ETAPS'06), Vienna, Austria, (2006), 82-96.