A PERVASIVE IDENTIFICATION AND ADAPTATION SYSTEM FOR THE SMART HOUSE

Paulo F. F. Rosa[1], Sandro S. Lima[1], Wagner T. Botelho[1], Alexandre F. Nascimento[2] and Max Silva Alaluna[1]

[1]*Military Institute of Engineering,Systems Eng. Dept, Rio de Janeiro, Brazil;* [2] *6th Military Region, Salvador, Brazil*

Abstract: The smart house is a system composed of several cooperating agents, which make it possible to act autonomously in performing tasks, that in conventional houses are done by their inhabitants. The current work proposes a smart house capable: (a) of identifying the inhabitants in the house, (b) to adapt itself to the inhabitants preferences, as well as to modify itself to changes on these preferences, and (c)to maintain optimal levels of safety, energy saving and comfort. This work develops an identification and adaptation system for the smart house, based on a multiagent methodology. This approach simplifies the implementation of the communication policy among the agents; besides, it gives more flexibility to the system, by delegating to each agent a specific job in the house. New jobs are added to the system by simply developing the necessary agents to execute them. The system is built through the definition of all the tasks for the house; at this point, the agents and their relationships are defined, in order to properly execute the tasks. This system uses a pervasive human footstep sensor capable of recognizing the inhabitants of the house, through their weights and the characteristics of their locomotion. A neural network is used to learn and adapt to the daily habits of each individual, as well as to properly negotiate the changes in these patterns of behavior. In order to validate our idea, experiments were done with a room's lighting system, the results of which show a very reliable system, capable to perform personalized actions in the house.

Key words: smart house; domotics; pervasive computing; footstep sensor; sensor network; multiagent systems; neural network; consumer electronics.

1. INTRODUCTION

The smart house is a system composed of several cooperating agents that make it possible to act autonomously in performing tasks that in conventional houses are done by their inhabitants. The house is composed of several environments, in which local agents are used to perform cooperatively many tasks. Furthermore, two global agents are defined: *water* and *energy agents,* which represent the main resources in the house. The current work proposes a smart house capable: (a) of identifying the inhabitants in the house, (b) to adapt itself to the inhabitants preferences, as well as to modify itself to changes on these preferences, and (c) to maintain optimal levels of safety, energy saving and comfort.

This work develops an identification and adaptation system for the smart house, based on a multiagent methodology. This approach simplifies the implementation of the communication policy among the agents; besides, it gives more flexibility to the system, by delegating to each agent a specific job in the house. New jobs are added to the system by simply developing the necessary agents to execute them. There are several methodologies that facilitate the construction of multiagent systems, for example: MaSE[1,2], Gaia[3,4], Message/UML[5,6], Prometheus[7–10] and Tropos[11,12]. This work combines the MaSE methodology and UML (Unified Modelling Language). This approach describes all the available services in high level just to broke them down in more details; so, all the necessary agents to the construction of the system can be identified, as well as the their relationships to each other.

The MaSE methodology uses graphic models and consists in two phases: analysis and design. The analysis identifies subgoals necessary to accomplish the main goals - to each goal is asserted a rule and a group of agent class. This phase includes three steps: capturing goals, applying use cases and refining roles. Initially, we define and structure the main goal as a collection of subgoals to be accomplished. Next, use cases translates the goals into rules and association of tasks. Finally, the roles are defined using sequence diagrams, it is necessary to assure that all roles had been identified; so, we will be able to develop the tasks that define their behavior and the communication pattern. In the design phase, the system is implemented, indeed. It consists on three steps: creating agent classes, constructing conversations and assembling agent classes. The first step consists on the creation of the agent's class diagrams that describe the system as a whole, showing its functioning and relationships. The second step defines a coordination protocol for the agents. The third step is necessary to define the agent architecture and its components. The final step of the design phase is the system design that involves building a deployment diagram which specifies the locations of the agents within a system. In the next section, the system is explained in more details.

2. THE SMART HOUSE PROJECT

In order to develop the system, it is necessary to identify all of its performable tasks. The identification of these tasks is facilitated through use case diagrams that depict a general view of all the available services in the house. However, this diagram presents no further details regarding how and by whom such tasks will be performed. At this phase of the project, the house might be able to: identify its inhabitants; to adapt the room to the preferences of its occupants; to learn new preferences delivered by the residents; to react to external influences disturbing the environment settings; and, to detect toxic gases and fire. From now on, we present the description of the tasks performed by the house.

Identification: A person entering in a room, activates the footstep sensor[13,14]. This event starts data acquisition by the sensor, that will inputs to the footstep algorithms, in order to determine the person's walking features. An ART1 (Adaptive Resonance Theory) neural network uses the results from the footstep algorithm to learn and associate the way of walking to a specific individual. Ergonomic studies[15] point to the uniqueness of footstep features within a group of persons. This system is capable to recognize new users in real time, as well as to detect slight variations on the locomotion patterns of the usual inhabitants.

Adaptation: after receiving the identification from the resident, the room makes a research on the resident's preferences and sends them to the agents that control the services supplied by the house. As soon as the tasks are concluded, the agents send a warning to the room; finally, the service will be finished when all the agents conclude their jobs.

Lighting settings: unsatisfied with the luminosity, the resident will be able to adjust the dimmer, causing the system to do a new reading of a light sensor, and the new preference of the resident is updated.

Temperature settings: when the resident modifies the temperature of a room, the system updates his/her preference in the profile databank.

Lighting compensation: weather variations can influence the luminosity of a room. This event is not done by the resident (resident's interference are done using the dimmer), so the system compensates the luminosity changes accordingly to the person's profile.

Gas protection: gas sensors are constantly monitoring the rooms. When dangerous levels are reached, the gas supply is interrupted, the alarm is turned on and the exhaustion begins working, until normality is reached.

The details of the services previously presented can be graphically represented, through activity diagrams. In these diagrams, we can view all the stages of a service, the decision points and the alterations that occur in the elements of the system, during the task. Figure 1 and Figure 2(a) shows the use cases diagrams and the activities diagram for the use case Identification. From this activity diagram we can see that we need to implement 2 (two) agents: the *Identification Agent* and the *Security Agent.*

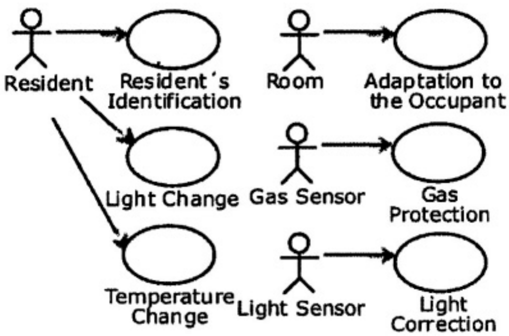

Figure 1. The use cases

The agents that are necessary to achieve the required functionality are obtained through the analysis of all the diagrams of the system activity, and will be addressed in the next section.

2.1 Defining the agents

Is this section, we present the agents that execute the system's tasks. The *id agent* recognizes the resident, permitting the house to act in a personalized way. The *room agent* receives the resident's identification and starts the adaptation process, by sending messages to the agents that control the equipments in the room. The *temperature agent* is responsible for keeping the room temperature accordingly to its occupant's preference, and also for updating the person changing profile. The *lighting agent* and the *noise agent* act just like the temperature agent, but in the illumination and noise fields. The *security agent* acts together with the id agent protecting the house against burglars; it is also responsible for detecting toxic gases and fire. The *electricity* and *water agents* control the use of these resources in the house; they can also be used in consumption's control policies. In order to allow for personalized actions, the room agent adapts the room to the resident's preferences, after his identification. These information are supplied by the *resident agent* [see Figure 2(b)]. We can verify that there is a group of preferences for each room (i.e., temperature, light and noise levels), but due to the system's construction methodology, new preferences can be included as the system grows.

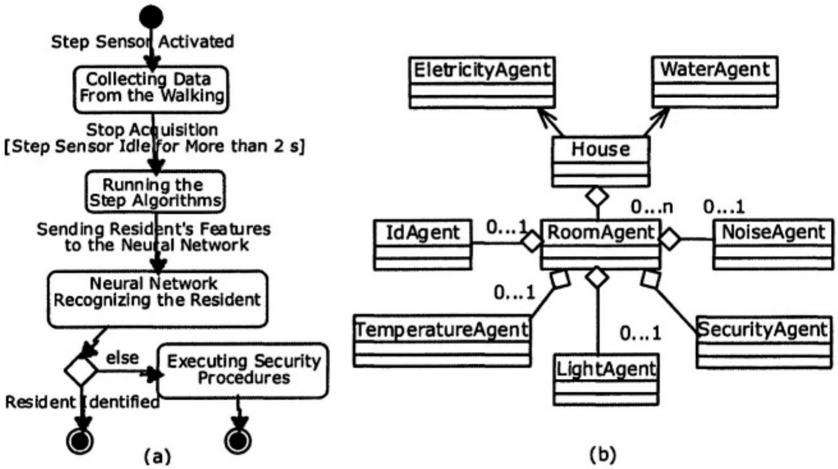

Figure 2. (a) Activity diagram of the resident identification. (b) The intelligent house agent

The agents that adapt the room to its occupant make sensors reading to define their actions, such as to send appropriate commands to the actuators. It can be verified that the system interacts with several equipments, such as: dimmers, thermostats in the cooler, sensors and home appliances in general. These equipments usually do none direct communication with a computer; consequently, it is necessary to develop an *Interface Class,* that will be composed of devices, such as: acquisition data boards and X-10 standard control panels.

After defining the agents of the house and their roles in the execution of the tasks, the next section will present the agents' control policies.

2.2 The control policy

The policy adopted by the system, for the communication among the agents, considers that the resident is identified by the id agent, when the resident agent enters a room in the house. The id agent informs the room the identity of the resident. The room agent gets the information about the residents' preferences with the resident agent, so it will be able to adapt the room. After getting the residents' preferences, the room agent asks the lighting, temperature and noise agents to allocate the necessary resources to the levels accepted by the resident.

When some non-identified person enters a room, the security agent is informed and it takes the proper action against the burglar. Besides, this agent, in conjunction with the temperature agent, is responsible for protecting the house against fire. As an example of the control policies, Figure 3 shows the communication among agents during the adaptation of a room.

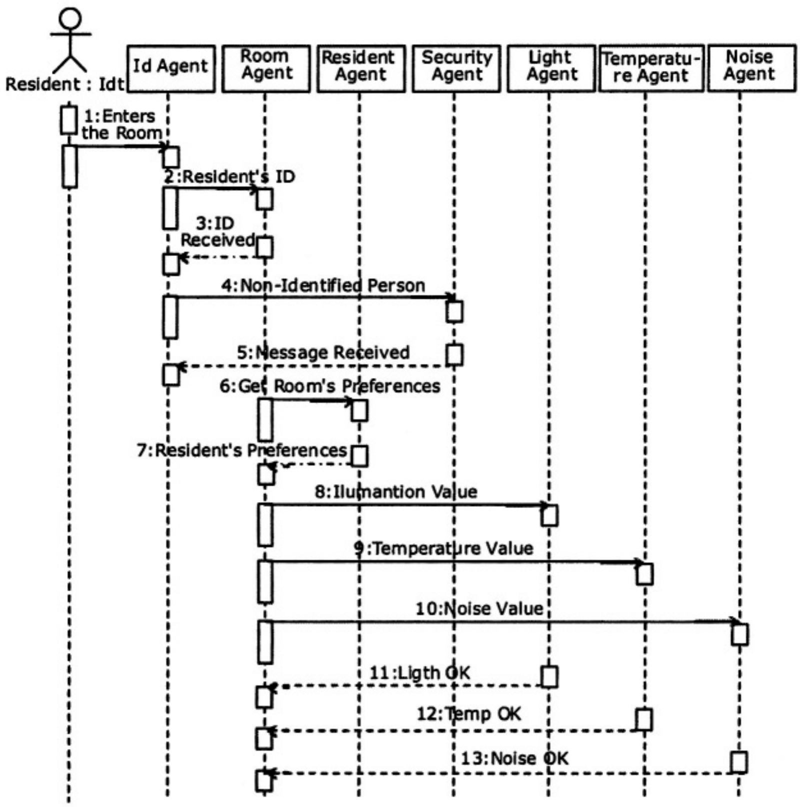

Figure 3. Room adaptation.

Finally, the house has to identify the residents, in order to act in a personalized way. In the next section, we will present, in detail, the id agent. Based on its work, every other agent in the house performs their job.

2.3 The id agent

The distinction of the persons living in the house might be done in the most natural and non-invasive way, in order to take into account concerns such as privacy and system's user friendlessness. We use a footstep sensor developed by Nascimento[13,15], which is capable to identify the residents on a house based on the features of their walking. The id agent is formed by three parts: (1) a sensor network; (2) a footstep algorithm; and (3) a neural network for classification and identification.

The sensor network collects data, in order to obtain the physical features to uniquely identify the resident, such as: f (footstep frequency), p (person's weight), δ_d (angle of the right foot), δ_e (angle of the left foot) and s (footstep length).

The sensor network is composed of load cells, the locations of which are described with (X, Y) cartesian coordinates. The load cells are disposed such as we can see in Figure 4. The distance d between the sensors has to be chosen carefully, since its value should permit the correct identification of a resident with a minimum data amount and computational effort. The value $d = 2cm$ was chosen, because it provided the best results with the minimum efforts, considering an adult's foot characteristics. In order to track the residents inside the house, a sensor network is placed in the transition zones for each room and in the house's entrances. The sensor network is able to identify several residents at a time, due to the fact that for a distance between two footsteps greater than $90cm$, they are considered to be from different persons. Footsteps laying within $90cm$ from each other might belong to a single person, and are known as *correlate group*.

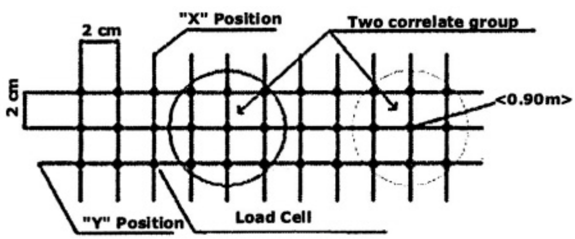

Figure 4. The sensor net.

The second part of the id agent is composed of the footstep algorithms: (1) footstep angle algorithm; and (2) the footstep frequency algorithm. In this paper, we work with variations of the the algorithms presented in works[13,15]. These algorithms receive the data collected by the sensor network and calculate the walking features of the resident.

The parameter p is an output from the load cell. The parameters δ_d and δ_e are determined by the footstep position and the direction of the person's locomotion. The parameter s is obtained by the distance between the right and left feet. We can observe that, as we walk, our feet accomplishes three phases: the support phase, the sustaining phase, and the impulse phase[1]. Let T_a, T_s and T_i be the time period for the support, sustain and impulse phases, respectively. Experimentally, it was observed that these phases demand the same amount of time; i.e., $Ta = Ts = Ti = T/3$. In this work, we considered these three phases with length $D=3$, where D is the footstep length. To calculate the footstep angle, we must find the last and the first sensors, that have been activated in that footstep. Then, we calculate the middle point between them, and the supporting points that are located in the first and second parts of the same footstep. The angle of the foot (right or left) is discovered using two equations for concurrent straight lines in any point. Then, we find the points (A, P_1 and P_2) for the footstep. The footstep direction is given by the line between the points P_2 and P_1. The straight line between the points A and A' shows the direction of the movement. The angle δ_d is given by the direction of movement and the footstep direction of the right foot. The data acquisition system works with a data structure (X, Y, p, T, E), where: (X, Y) are the cartesian coordinates of the footstep sensor; P is the weight; T indicates the time, in which the sensor was activated or deactivated (0-deactivated and 1-activated). In order to obtain the frequency f, the algorithm calculates the time interval between the first sensors activated on footstep i and $i+1$. Furthermore, to determine the frequency, the algorithm uses the structure (X, Y, T), that can find the coordinates and the time of activation of the first sensor in each footstep. The final value of the frequency will be the mean value of the frequencies calculated in this process. The characteristics of walking are presented in Figure 5.

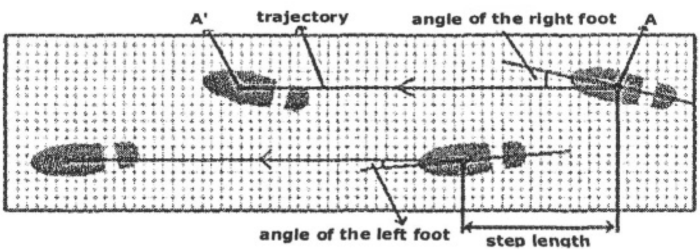

Figure 5. The resident's walking features.

The neural network is used to identify the home's resident using his/her walking features calculated by footstep algorithms. The ART1 neural

[1] Here is a brief explanation of the three phases of a person's walk: *(a) Support phase:* when the swing foot reachs the ground; *(b) Sustaining phase:* the entire foot is placed on the floor, preparing for an impulse; and *(c) Impulse phase:* the foot propels the body forward and the leg loses contact with the ground.

network is used because its training algorithm is non-supervised, which eliminates the need of previous presentation of the residents' walking patterns; and also, for being capable to recognize a resident that presents variations in his/her walking.

In order to validate the id agent, we have to show that it is capable of recognizing new individuals (as well as, being aware of that in future), even if they present variations of their walking features. The footstep patterns of some individuals were collected as experimental data, using a plane ground surface covered with paper. So, during the people's walk, the footsteps (the shoes were marked with ink) were registered in the paper, allowing us to collect the data needed for the footstep algorithms, that is summarized in Table 1.

Table 1. Features extracted from 10 persons.

Individual	δ_e [°]	δ_d [°]	s[cm]
1	-27.6 ± 1.7	22.7 ± 1.8	52.7 ± 1.3
2	-35.8 ± 2.4	28.2 ± 1.2	55.7 ± 1.1
3	-34.7 ± 1.9	28.4 ± 3.1	78.1 ± 1.7
4	-26.6 ± 2.1	25.1 ± 2.3	65.0 ± 1.6
5	-24.2 ± 1.7	21.4 ± 1.5	70.2 ± 2.1
6	-42.4 ± 2.7	31.5 ± 2.6	59.6 ± 1.9
7	-31.7 ± 2.6	23.8 ± 1.8	72.5 ± 1.5
8	-16.5 ± 2.4	12.5 ± 3.2	68.0 ± 1.0
9	-21.2 ± 2.9	18.8 ± 2.7	59.5 ± 1.8
10	-33.2 ± 1.6	30.3 ± 2.4	75.4 ± 1.5

The ART1 neural network uses binary input, so we have to make some considerations. The input variable p ranges from 20*kg* to 147*kg*, so it is represented using a vector of 7 bits. The input variable f ranges from 1*Hz* to 2.2*Hz*. If the precision used is 0.10*Hz*, then we will have 13 possible frequency levels, so it is represented with a vector of 4 bits. The input variables δ_d, δ_e and s follow no specific pattern. From the data collected, we can see that the angle of the left foot presents a variation of -25.9° (**$-42.4° < \delta_e < -16.5°$**) and the angle of the right foot presents a variation of 19.0° (**$+12.5° < \delta_d < +31.5°$**). If the precision adopted is 0.1°, we need a vector of 9 bits for the angle of the left foot, and a vector of 8 bits for the angle of the right foot. The footstep length s presented a variation of 25.4*cm* (52.7*cm* < *s* < 78.1*cm*). Consequently, an 8-bit vector is enough, for a 0.1cm precision. If the output variable Idt can be represented by a vector of 4 bits, we can identify up to 16 people - a more than reasonable number of persons in a house.

Let us suppose the existence of a house with the residents with the features displayed in Table 2.

Table 2. **Residents features.**

Idt	*f [Hz]*	*P [Kg]*	δ_e *[°]*	δ_d *[°]*	*s [cm]*
1	2.2	80	-27.6	22.7	52.7
2	1.2	65	-35.8	28.2	55.7
3	2.0	45	-34.7	28.4	78.1
4	1.7	50	-26.6	25.1	65.0
5	1.8	70	-24.2	21.4	70.2
6	1.4	75	-42.4	31.5	59.6
7	2.1	95	-31.7	23.8	72.5
8	1.6	85	-16.5	12.5	68.2

Since the ART1 neural network works with binary data, we convert the features in Table 2, and each person is now represented by a 36 bits vector. This vector contains all the studied features and will be the input for the neural network and are shown in Table 3.

Table 3. Residents features input for the ART1 neural network.

Idt	*f [Hz]*	*p [Kg]*	δ_e *[°]*	δ_d *[°]*	*s [cm]*
1	1100_2	0111100_2	001101111_2	01100110_2	00000000_2
2	0010_2	0101101_2	011000001_2	10011101_2	00011110_2
3	1010_2	0011001_2	010110110_2	10011111_2	11111110_2
4	0111_2	0011110_2	001100101_2	01111110_2	01111011_2
5	1000_2	0110010_2	001001101_2	01011001_2	10101111_2
6	0100_2	0110111_2	100000011_2	10111110_2	01000101_2
7	1011_2	1001011_2	010011000_2	01110001_2	11000110_2
8	0110_2	1000001_2	000000000_2	00000000_2	10011011_2

For a better visualization, each individual pattern will be represented by a block formed by small white and black squares. A white square represents a bit 1 (one) an a black square a bit 0 (zero). The blocks should be read from the top to the bottom, from left to right.

First, we will show that the id agent is capable of recognizing the residents of the house; so, we developed a matlab function that implements the ART1 neural network. This function takes as parameters the output of the footstep algorithms. Figure 6 shows the final configuration of the neural network after the residents features being presented; in the top of the figure, we can see the patterns from each resident and in the bottom we can see the clusters formed for each pattern recognized by the agent.

Now, we have to show that the id agent is capable of identifying a resident, even if there are variations in his/her walking features. These variations should be saved in the resident's cluster. Suppose, for example, that the resident 1 is carrying a bag, so his/her weight changes from 80 to **90*kg* = 1000110_2** and his/her footstep length, s, goes from 52.7*cm* to **54.0 *cm* = 00001101_2.** Imagine also that resident 7 is going on a diet, and now, s/he is weighting $\mathbf{82}\boldsymbol{kg} = \mathbf{0111110_2}$ and his/her footstep frequency now is $\mathbf{1.7}\boldsymbol{Hz} = \mathbf{0111_2}$.

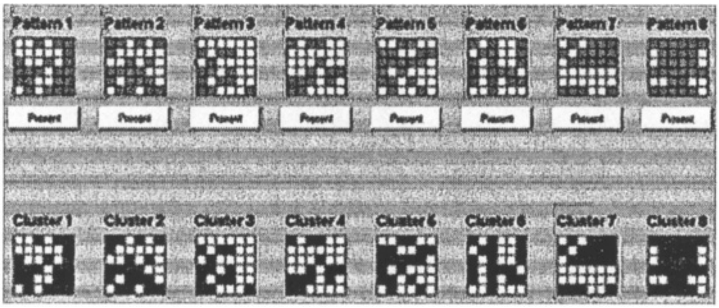

Figure 6. Recognition of the residents.

Figure 7 shows the final configuration of the neural network after the presentation of the new features of residents 1 and 7, the ART1 neural network has learned the new patterns of the residents and saved them to their clusters.

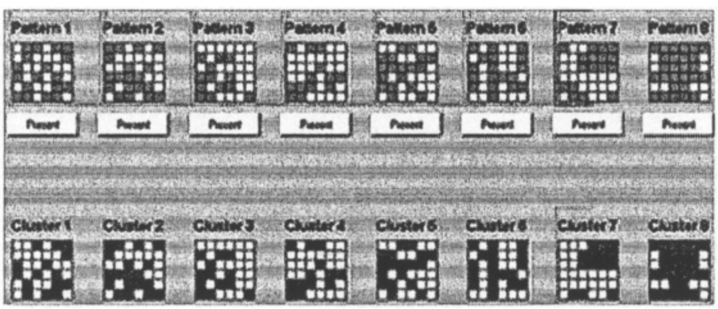

Figure 7. Recognition of the residents with variation on walking patterns in residents 1 and 7.

We have proved that the id agent accomplished its role very well, being able to identify new users, creating new clusters for them; and recognizing variations on patterns already presented to the network, making the necessary changes on the clusters in order to learn these new characteristics.

In the next section, we introduce the lighting agent, which uses the information provided by the resident agent to adjust the lights of a room based on the occupant's preferences.

3. LIGHTING AGENT

The lighting agent that we propose sets the level of the lights of each room based on the preferences of the resident recognized by the id agent. The system consists of a photoresistor and a dimmer, which can be operated manually or through a microcontroller. Hereafter, we make a description of the learning process. As soon as an identified resident enters for the first time in a room, and stays there for 10 seconds at least, the system should keep the last value informed by the photoresistor. The system will track the person all over the house for a certain time, in order to pick up a report of his/her light setting preferences. This period of 10*s* is enough to the person to change the light level of a room. For the very first 3 times of a person in a room, the illumination will be activated in 60% of its maximum value and it will be reduced until the user changes the position of the dimmer. Besides, if a person increased the dimmer's set 3 (three) times, the next time the light level will be set in 100%.

The total luminous flux of a room is given by: $\Phi T = \Phi 1 + \Phi 2 + \Phi 3$, where they are, respectively: the flux of the controlled room, the flux from the natural source and the flux from neighboring rooms.

To test our lighting agent, we modeled a room using the matlab. In the simulation we considered resident's illumination level preference of 3.780 lumens, and that the lamps installed in the room are capable to produce a maximum luminous flux of 5.400 lumens. During all day, variations of the luminous flux occurred in the room, see Table 4.

Table 4. [Variation of the Liminous Flux of a Room During a Day.]

ΦTA	*70% PmaxA=3.780lumens*	*Constant*
ΦT1A	(5.400x*NdA*) / 20	*NdA*-dimmer's position A
ΦT2A	0	$0h < t \le 7h$
	[((1000x17) - (tx1000)) / 3]	$7h < t < 10h$
	1000	$10h \le t \le 14h$
	[((1000x17) - (tx1000)) / 3]	$14h < t < 17h$
	0	$17h \le t \le 0h$
ΦT3A	0	$0h < t \le 7h$
	[(120xt – 7x 120)/3] + 100	$7h < t < 10h$
	220 sin 1.57xt + 220	$10h \le t \le 14h$
	[((120x7)-(tx120))/3]+100	$14h < t < 17h$
	0	$17h \le t \le 0h$

The lighting agent has to maintain the luminous flux of the room constant during all day, despite the action of the natural source and neighbor rooms. We can see, in Figure 8, that the agent reduces or increases the lights of the room according to the external sources of light.

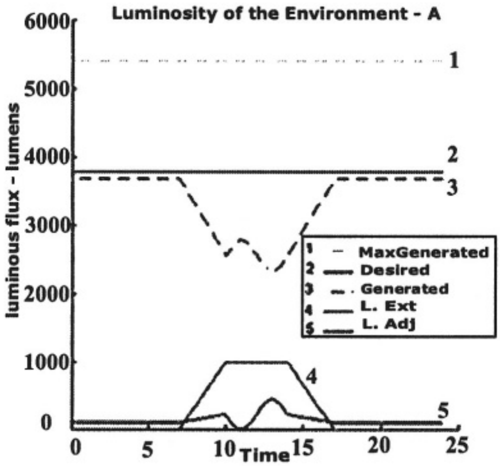

Figure 8. [Luminous flux of the room.]

Using fluorescent lamps in the room we achieved an energy saving of 707 *Wh* in 24 hours, which means a reduction of 36.8%. Therefore, the lighting agent presented here accomplished its goal of recognizing the individual, and most of all, the agent saved energy keeping the comfort levels stable, what is one of the objectives of the smart home.

4. FINAL COMMENTS

The current work proposed a smart house capable of identifying the inhabitants in the house, adapting itself to the inhabitants daily preferences, as well as to modify itself to changes on these preferences, and maintaining optimal levels of safety, energy saving and comfort. This work developed an identification and adaptation system for the smart house, based on a multiagent methodology. This approach simplifies the implementation of the communication policy among the agents; besides, it gives more flexibility to the system, by delegating to each agent a specific job in the house. The system uses a pervasive human footstep sensor capable of recognizing the inhabitants of the house, through their weights and the characteristics of their locomotion. A neural network is used to learn and adapt to the daily habits of each individual, as well as to properly negotiate the changes in these patterns of behavior. In order to validate our idea, experiments were done with a room light system, the results of which show a very reliable system, capable to perform personalized actions in the house. As a next approach to this research, we intend to build a prototype for the entire system and implement features of other multiagent modeling methodologies to optimize the performance of our smart house project.

REFERENCES

1. S. A. DeLoach. Analysis and design using MaSE and agentTool, in Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001), 2001.
2. S. A. DeLoach, M. F. Wood, and C. H. Sparkman, Multiagent systems engineering, International Journal of Software Engineering and Knowledge Engineering, 11(3): 231-258, 2001.
3. M. Wooldridge, N.R. Jennings, and D. Kinny, A methodology for agent-oriented analysis and design, In Proceedings of the third international conference on Autonomous Agents (Agents-99), Seattle, WA, May 1999, ACM.
4. M. Wooldridge, N.R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems, 3(3), 2000.
5. G. Caire, F. Leal, P. Chainho, R. Evans, F.G. Jorge, G. Juan Pavon, P. Kearney, J. Stark, and P Massonet. Project p907, deliverable 3, Methodology for agent-oriented software engineering. Technical Information Final version, European Institute for Research and Strategic Studies in Telecommunications (EURESCOM), 09 2001.
6. Giovanni Caire, Francisco Leal, Paulo Chainho, Richard Evans, Francisco Garijo, Jorge Gomez, Juan Pavon, Paul Kearney, Jamie Stark, and Philippe Massonet, Agent oriented analysis using MESSAGE/UML, In Michael Wooldridge, Paolo Ciancarini, and Gerhard Weiss, editors, Second International Workshop on Agent-Oriented Software Engineering (AOSE-2001), pages 101-108, 2001.
7. L. Padgham and M. Winiko, Prometheus: A methodology for developing intelligent agents, In 3rd International Workshop on Agent-Oriented Software Engineering, July 2002.
8. L. Padgham and M. Winiko, Prometheus: A pragmatic methodology for engineering intelligent agents, In Proceedings of the OOPSLA 2002 Workshop on Agent Oriented Methodologies, pages 97108, Seattle, November 2002.
9. L. Padgham and M. Winiko, Prometheus: Engineering intelligent agents, tutorial notes, available from the authors, October 2002.
10. L. Padgham and M. Winiko, Prometheus: A brief summary, technical note, available from the authors, January 2003.
11. F. Giunchiglia, J. Mylopoulos and A. Perini, The Tropos software development methodology: Processes, Models and Diagrams, in Third International Workshop on Agent-Oriented Software Engineering, July 2002.
12. P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, Troops: An agent-oriented software development methodology. Technical Report DIT-02-0015, University of Trento, Department of Information and Communication Technology, 2002.
13. A. F. Nascimento, Dynamic system of Residential Automation, Master Thesis in Systems and Computation, Military Institute of Engineering - IME, Rio de Janeiro, Brazil, 2002.
14. P. F. F. Rosa, A. F. Nascimento, S. S. Lima, "Domotic: A Pervasive Identification System for the Smart House". World Congress on Engineering and Technology Education, 2004 WCETE, pp. 1233-1238.
15. R. J. Orr, and G. D. Abowd, The Smart Floor: A Mechanism for Natural User Identification and Tracking, in Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems: Extended Abstracts, ACM Press, New York (2000), pp. 275-276.