Enhancing the Human-System comprehension in a Smart Home System using Pervasive Computing

Celine Maria Amrita A. S., Harini S., Kiran Sharmilee D, S. Swarna Parvathi

Department of Information Technology, Sri Venkateswara College of Engineering
Pennalur, Sriperumbudur, Kancheepuram District, Tamil Nadu, India

ABSTRACT
In a home environment, human actions consist mainly of routines which are performed at approximately the same time every day, and one routine is broken only to be replaced by another. This paper is based on a pervasive interaction with the inhabitants of a house to provide an enhanced Smart Home experience. This paper proposes a probabilistic model to manipulate the environmental information from sensors and identify the actions of the household inhabitants, by exploiting the fact that household activities encompass routines. This paper also shows the simulation of a smart home system, which uses this probabilistic model to adapt to one's habits, by identifying one's intensions and accordingly transforming the household environment to support the user's intended actions.

1 INTRODUCTION
A smart home system integrates the electrical devices in a house and allows control of the devices through a personal computer or remote control. The techniques employed in home automation include control of domestic activities such as home entertainment systems, lighting system, pet feeding and changing the ambience for different events such as parties or dinners. The devices may be controlled via the computer network and may be accessed remotely from the internet. By integrating the concepts of information technology with the home environment, systems and appliances are able to interact in a seamless manner which results in convenience; energy efficiency and safety benefits.

The existing smart home systems depend on explicit commands from the inhabitants. It ignores the fact that one can perform routine actions in their day-to-day life. This can be overcome by using the ideas of pervasive computing in order to predict the actions based on information from various kinds of sensors and thereby eliminating the need for an explicit human-system interface, such as voice commands or button presses from the household inhabitants.

The system being developed is a simulation of the home environment in 2D, which graphically depicts the actions that take place around the house, including user's action and the resulting transformation of the home environment. The system uses a probabilistic algorithm to identify user's actions. In order to build a smart home system, there is a need for a Home Service Gateway to interconnect the household devices to the controlling system and for this purpose we use the Open Service Gateway Initiative OSGi Framework. There is also a need for sensors and controllers, for which we use virtual devices which are OSGi bundles that interact with the OSGi framework. Sensors to collect information about the household environment and communicate this information to the framework; and controllers which pass on control information from the framework to various household devices.

2 BACKGROUND
A background on the project can be obtained by understanding the following terminologies and concepts.

2.1 Context Information
The information collected from the sensors concerning the home environment is called context information. As there are various kinds of sensors which can provide information of the environment, such as position sensors, light intensity sensors, temperature sensors, etc. there are a variety of context information that are available . This context information is manipulated to understand the current situation, and thereby give the desired transformation of the home environment for the situation. Thus the smart home can become situation aware and provide a greater a degree of ease.

2.2 OSGi

A variety of household appliances, multimedia, communication and security protection devices are connected and controlled conveniently by a general remote control device which is called Home Service Gateway. Up to now, OSGi is the most popular service gateway, which is compatible with many existing standards. The home service gateway based on OSGi supports interconnection of a variety of information house hold appliances; it could autonomously construct a single dynamic Smart home network which contains a variety of appliances, communication devices and computers by the self-adapting mechanism of OSGi. The Open Services Gateway initiative framework is a module system and service platform for the Java programming language that implements a complete and dynamic component model. Applications or components in the form of bundles for deployment can be remotely installed, started, stopped, updated and uninstalled without requiring a reboot. More information on OSGi can be obtained from [4].

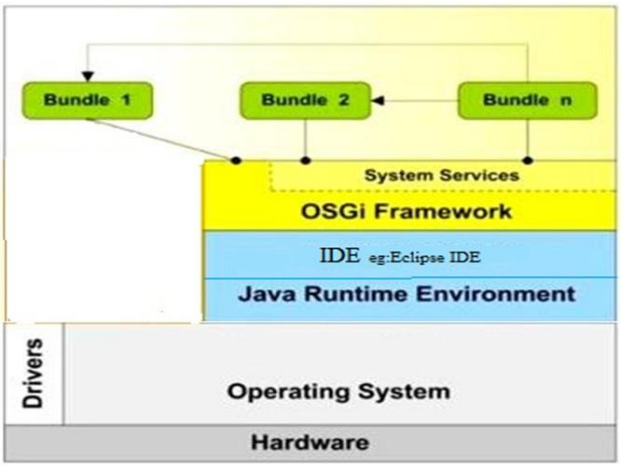

Fig. 1 OSGi Architecture

Any framework that implements the OSGi standard provides an environment for the modularization of applications into smaller bundles. The framework is conceptually divided into the following areas:

Bundles: Each bundle is a tightly-coupled, dynamically loadable collection of classes, jars, and configuration files that explicitly declare their external dependencies (if any). Bundles are normal jar components with extra manifest headers. A bundle is a group of Java classes and additional resources equipped with a detailed manifest MANIFEST.MF file on all its contents, as well as additional services needed to give the included group of Java classes more sophisticated behaviors, to the extent of deeming the entire aggregate a component.

Services: The services layer connects bundles in a dynamic way by offering a publish-find-bind model for plain old Java Interfaces POJI or Plain Old Java Objects POJO.

OSGi by itself is a specification for which several implementations are available. We are using the Knopflerfish open source OSGi implementation for the simulation environment as described in the next section.

2.3 Knowledge Networks

To make contextual information meaningful and useful, some tools must be made available to pervasive services, which can properly correlate and predigest contextual information so as to provide them with a higher level understanding of surrounding. One should consider the following:

1) The KN approach promotes a clear separation of concerns that can notably reduce the complexity of developing and maintaining services, and

2) In a distributed setting, KNs can take care of knowledge management duties that would have been otherwise replicated inside each service.

To avoid pervasive services to access and digest large amounts of data directly, a sort of "middle layer" must be placed in between the data sources and the services. Such a middle layer will be in charge of collecting data items, analysing them, and building a compact higher level view of the context. This layer could facilitate the aggregation of user profiles, possibly merging them with sensorial information, in order to provide situation-specific knowledge to services and to enable them to immediately act on this knowledge.

Let us assume that there are various kinds of "sensors" generating large amounts of independent atomic units of contextual information. We can call these as "knowledge atoms." The KN approach considers exploiting self-organization approaches to aggregate/correlate/prune such knowledge atoms to facilitate their exploitation by services. For deeper understanding of the Knowledge Networks, [1] can be referred to. This includes the following possible dimensions in which knowledge may be organized

***Semantic dimension*:** Knowledge atoms that are related to a situation relate to each other according to the concepts that are available or inferred from. e.g., a shared ontology.

Spatial dimension: Knowledge atoms that are related to a local fact can network to knowledge atoms at different locations.

Temporal dimension: Knowledge atoms express facts which have occurred (or are about to occur) at different times.

3 SIMULATION ENVIRONMENTS

The Simulation environment for the system under consideration is a 2D representation of the house plan chosen, shown in Fig. 5, which is created using a Java Applet along with the Applets of the sensor and controller OSGi services. The output so far contains a basic implementation of the algorithm and the various bundles for the sensors and controllers. The system mainly aims at controlling the lighting system i.e. lights that are ON/OFF and their intensity, based on the user's current activity. The environment can be described as follows:

3.1 Input and Output

Input: The system takes as input the various sensor values and produces as output the transformation in the home environment for the predicted action. These sensors are not real-time physical units, rather they are virtual devices. In our simulation environment, the virtual device bundles acquire the sensor values as text input to run the simulation and not real-time sensor values. Virtual devices are OSGi bundles that are capable of communicating with the framework and can be installed in information capable household appliances. However there is no difference between virtual bundle and real device bundle, meaning that the application code of a virtual device can be transplanted to real device with slight or no modification. Each of the sensor bundles runs a Java Applet to get the values of position, intensity, etc. Fig 4 shows the OSGi Position Sensor bundle displaying the plan and it take mouse click positions as the position of the inhabitant.

Output: The output from processing the sensor information includes:

- The predicted action that the user is intended to perform.
- The transformation of the home environment in the 2D plan, to show which lights are ON and their corresponding intensity value to suit the predicted action. Fig 4 shows the Lighting Simulation bundle which reflects the change in the lighting according to the detected action.

Feedback: The system requires that in case of a wrong prediction, the user makes manual adjustment of the lighting as required by the correct action. Fig 4 shows a Simulation Feedback bundle which provides for this option.

For example, in Table I when the action predicted by the algorithm is "Watches TV" the output will be such that the light identified as "lr01" is turned on and the intensity of the light is set to 60lux. This output is again an OSGi bundle running a Java Applet.

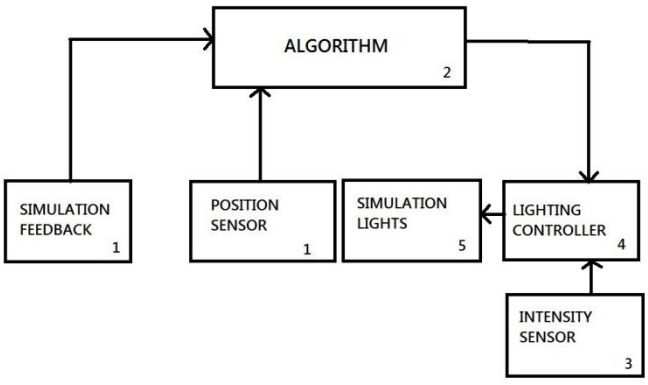

Fig. 2 Interaction between OSGi Bundles.

The Fig. 2 explains the interaction between the OSGi bundles. The number at the bottom of the boxes represent the order in which the bundles have to be run as one bundle depends on the existence of the other. The Simulation Feedback, Intensity Sensor and Position Sensor bundles do no depend on information from any other bundle. Algorithm requires the

position of user and the user's feedback and hence the Position Sensor and Simulation Feedback bundles. Similarly The Lighting Controller requires the existing light intensity from Intensity Sensor to sense the natural lighting of the room and also the detected action from the Algorithm bundle. The Lighting Controller sets the intensity of the lights in such a way that it complements the natural lighting of the room (This is a power saving option). The Simulation Lights provides a simulated display of the lights that are ON and their intensities.
The Fig. 4 shows the applets of various bundles running in the Knopflerfish OSGi desktop which includes the Position Sensor Bundle to input the position of the inhabitant based on a mouse click in the plan of the house displayed by it, the Light Intensity Sensor bundle which allows update of the available intensity, and the Algorithm Bundle which displays the position obtained(in the Fig. 4 the position clicked is 1322,408) from the Position Sensor and the predicted action(in the Fig. 4 the action is "User Reads") based on probability.

3.2 Algorithm

Fig 3. Shows the overall algorithm of the system. The concept of KNs is used to aggregate the actions of the inhabitants in the spatial and temporal dimensions. A predefined set of actions, that the inhabitant can possibly perform and for which the required lights and their intensities vary, is maintained. These actions are aggregated into groups of actions that are possible at a particular instance of time and those that are possible in a particular location in the house. Table 1 shows the list of actions that the user can possibly perform in the living room along with the corresponding light intensity for each action(for example: actions that can be performed in the living room at 17:00 hrs).

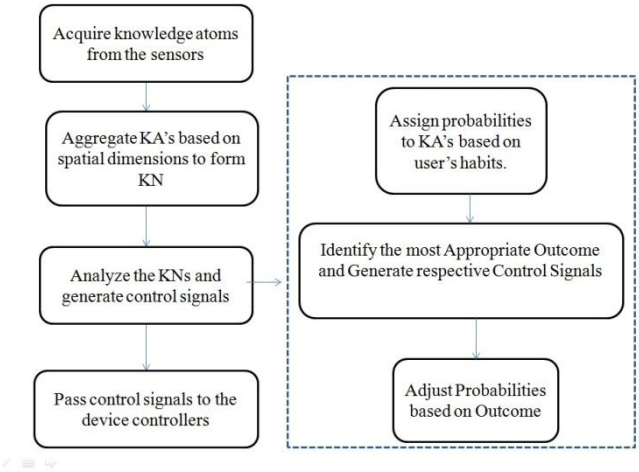

Fig. 3 The Overall Algorithm.

However there are many cases where more than one action is possible at a point in time and in a particular room, and in order to resolve to a single action we use a probabilistic model. According to this model each action is simply assigned a probability of occurring at a particular time. Consider the actions in Table I which shows the set of possible actions a user can perform in the Living room shown in Fig 6. Each of the possible actions is assigned a list of probabilities which tells the possibility of that action occurring at each point of time during the day. For example a 2-dimentional array (say P) that stores these probabilities is accessed as follows:

$$P[i][j] ,$$

where 'i' is the action,
'j' is the time of the day (in hours or minutes). And when there arise a conflict between actions, the action with the highest possibility of occurring at that time is chosen as the intended action.
The model allows the probabilities to be dynamic and thereby adapt to human habits of day-to-day living. To being, initial probabilities are assigned on a generic basis, obtaining the ADL (Activities of Daily Living) probabilities from any existing data set. The algorithm turns dynamic when the probabilities change ratios and the most probable action attains the highest value. This adjustment is made based on the success or failure of the previous prediction. In case of successful prediction the probabilities are maintained. However in case of a wrong prediction the probability of actions is re-assigned for future correctness.

For example, if at time T, the user is expected to perform the action "Reads", according to the existing (or initial) probabilities then P[i][T] is maximum, for 'i' as "Reads". If this prediction become incorrect, which the user can convey by changing the intensity manually to the desired value, the probability P[i][T] is reduced.

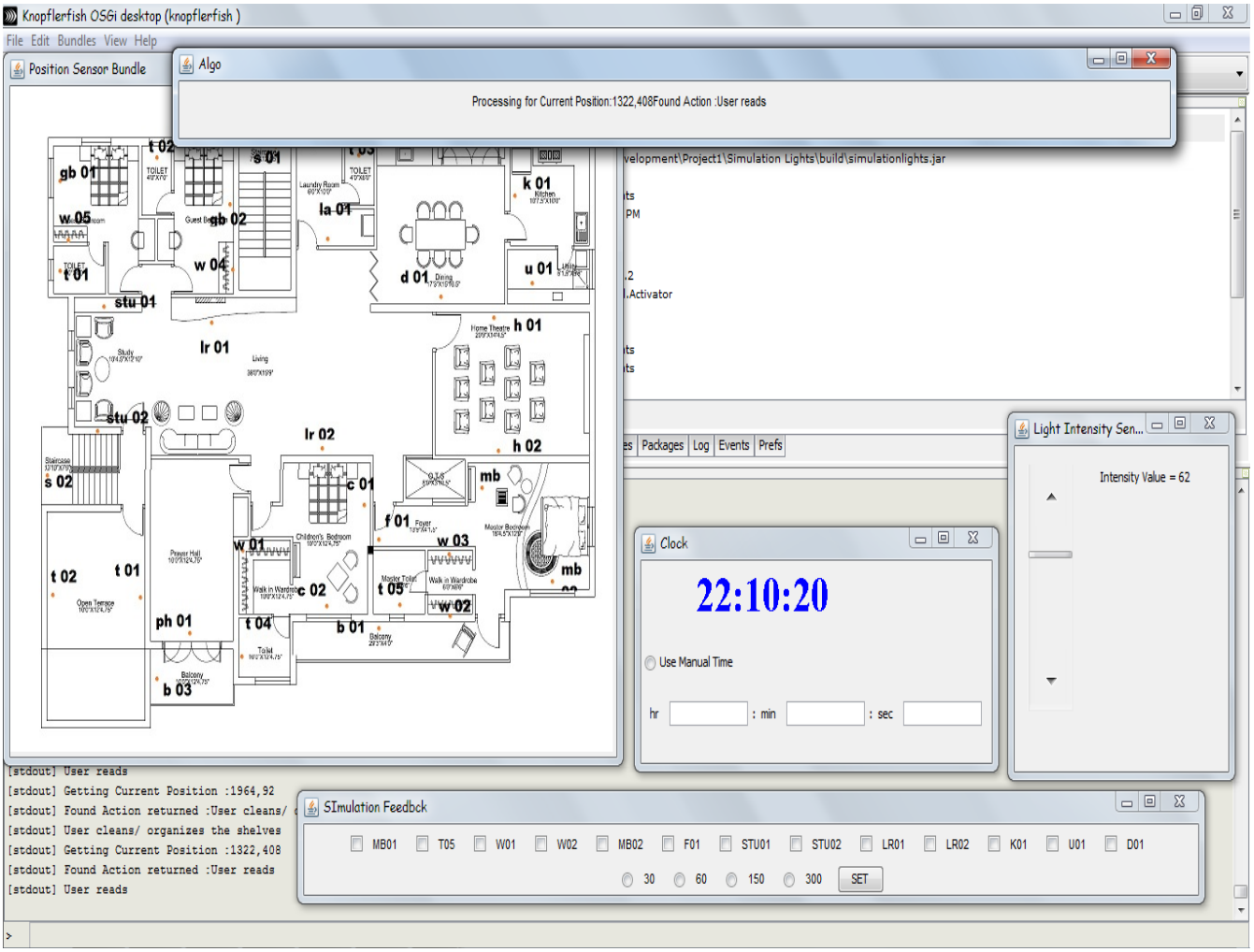

Fig. 4 The OSGi Desktop running the Position, Algo, Intensity Sensor, Lighting Controller Bundles.

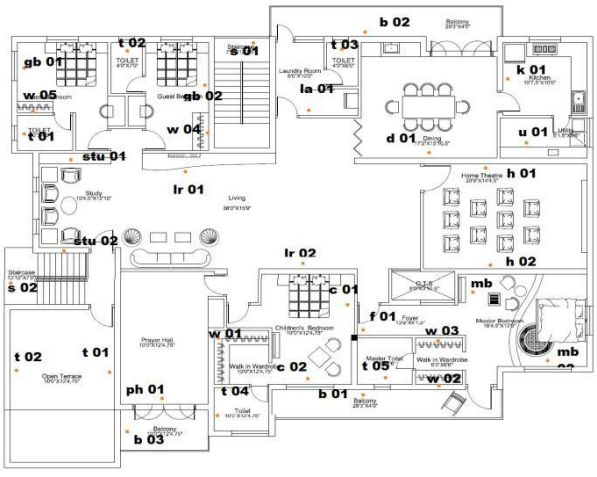

Fig. 5 Plan of the adopted house showing the placement and ID of lights.

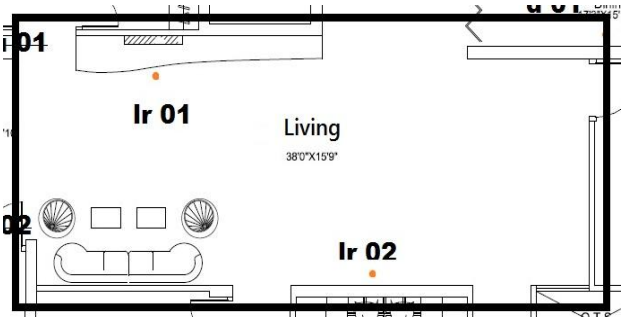

Fig. 6 Portion of Plan showing the living room.

Table 1. Table of Actions and Reactions

No.	Living Room		
	Action	**Required Lights ID**	**Required Intensity (lux)**
1	Watches TV	lr 02	60
2	Invites guests/ Conducts celebrations	lr01,lr02	300
3	Relaxes	lr02	30
4	Reads	lr 02	300
5	Other Activities	lr01,lr02	150

4 SIMULATION RESULTS

4.1 Algorithm Results

The simulation of smart home environment shows how the system adapts to the user's routine habits. The expected results are as follows:

Let us analyze the probabilities of user performing an action, for three consecutive days. Initially, as we have considered generic probabilities, the predictions tend to be erroneous. The Fig. 7 depicts a graph showing the probability distribution of the user's actions at time T for the first day.

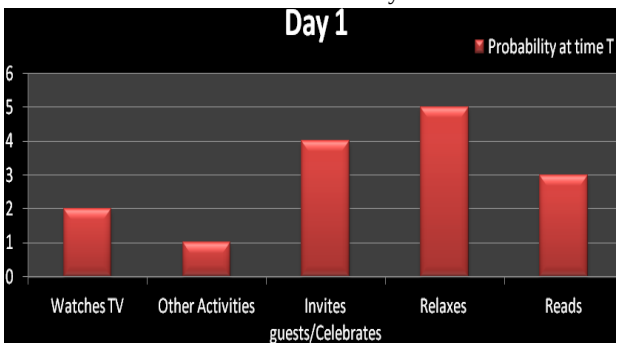

Fig. 7 Graph showing probabilities of users actions on Day 1.

According to the existing probabilities, the user was expected to perform the action "Relaxes". However, if it turns out to be wrong and the user performs "Invites Guests" instead, the system will modify the corresponding probabilities as shown in Fig. 8.

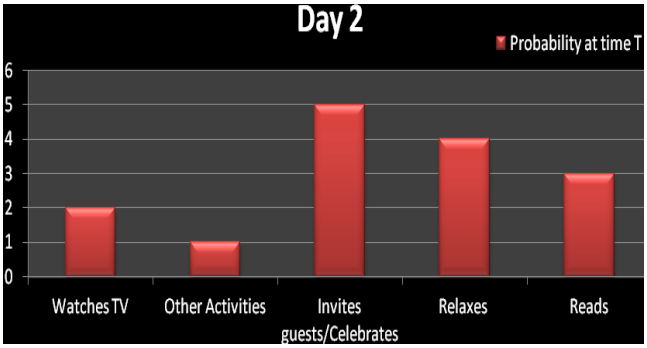

Fig. 8 Graph showing probabilities of users actions on Day 2.

Now the system assumes that the user performs "Invites Guests" on the third day and if the prediction turns out to be correct the probabilities remain unchanged from Day 2.

4.2 Simulation Output

User's movement across the rooms is tracked and based on the time and probability the prediction is made. The expected simulation output allows us to see the location of the user, and the changes that take place in the lighting system as and when the user performs a task. The user is allowed to show displeasure of the light setting made (i.e. In the case of a wrong prediction) and is allowed to manually change the lights that are ON or OFF and their intensities as required by his current action. This manual adjustment as mentioned in the algorithm, acts as the feedback to the system to adjust the probabilities for the next day's prediction. The Fig. 7 shows the final expected output of the system which will be displayed by the Simulation Lights Bundle and has the following features marked in the figure:

User Icon: This shows the current position of user in the house.
Move cursor: This is a cursor which allows us to make changes to the user's position in the house.
Light OFF: This is an example of a light turned off.
Light ON: This is an example of a light turned on.

This simple probabilistic model applied into the simulation environment will help the system adapt to activities that are routinely performed. The system can provide an enhanced experience if it keeps track of routines which can repeat itself later in the month, year or another year, such as seasonal events, work cycles, etc.

5 CONCLUSIONS AND FUTURE WORK

The proposed smart home system which uses probabilistic model to predict user's actions based on routine work is thus an example where pervasive computing can be applied in real time scenario. The information from the sensors is manipulated to predict the user's intended actions effectively. The core of the probabilistic model contained in the algorithm depicts a unique way of applying pervasiveness. The smart home system finds it's usefulness in many ways taking luxury to the next level. This system can be extended for the activities that can be performed on a monthly or yearly basis by including appropriate probability tables. Additional pervasiveness can be introduced into several other activities of daily living by controlling various other household appliances in a way similar to the lighting system. Another proposal would be to use an algorithm to generate the probability density function for the probabilities and thereby reducing memory requirements. The system implemented can also be improvised to support generic house layouts by integrating with legacy systems that create the design of the house.

6 REFERENCES

[1] Nicola Bicocchi, Marco Mamei,Rico Kusber,Franco Zambonelli *"Self Organized Data Ecologies For Pervasive Situation Aware Services- the Knowledge Network Approach"* IEEE Transactions On Systems, Man, And Cybernetics—part A: Systems And Humans, Vol. 40, No. 4, July 2010.

[2] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, "*Inferring Activities from Interactions with Objects"*, IEEE Pervasive Computing, 3(4):5057, 2004.

[3] C.Chong, S.Kumar, *"Sensor Networks: Evolution, Opportunities, Challenges"*, Proceedings of the IEEE, 91(8): 1247 - 1256, 2003.

[4] Zhang Lei, Suo Yue, Chen Yu, Shi Yuanchun *"SHSim: An OSGI-based smart home simulator"* Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference on 5-6 July 2010.

[5] J. Serrat, J. Serrano, J. Justo, R. Marín, A. Galis, K. Yang, D Raz, Efstathios D. Sykas, *"An Approach to Context AwareServices"*, NOMS2004, Seoul, South Korea, 2004.

[6] J. Hightower, G. Borriello, *"Location Systems for Ubiquitous Computing"*, IEEE Computer, 34(8): 57-66, 2001

[7] Myunggwon Hwang, Pankoo Kim, Dongjin Choi et al Information *"Retrieval Techniques to Grasp User Intention in Pervasive Computing Environment"* Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011

[8] Yunwei Dong, Bo Zhang, Kang Dong *"An Integrated PLC Smart Home System in Pervasive Computing Environment"*, Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing.