Agent Based Smart House Platform with Affective Control

Kuderna-Iulian Benţa
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
G.Bariţiu 25-26
+40-264-401-226
Iulian.Benta@com.utcluj.ro

Amalia Hoszu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
G.Bariţiu 25-26
+40-364-267-232
hoszu_amalia@yahoo.com

Lucia Văcariu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
G.Bariţiu 25-26
+40-264-401-477
Lucia.Vacariu@cs.utcluj.ro

Octavian Creţ
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
G.Bariţiu 25-26
+40-264-401-477
Octavian.Cret@cs.utcluj.ro

ABSTRACT
In this paper, we describe our work in developing an agent based smart house platform using TAOM4E development methodology and the JADE-platform with the Jadex-extension. In order to remotely monitor its functions we implemented a Jade-Leap mobile application. We developed a central ontology for context representation and the support for behavior changes according to user's affective feedback. We also emphasize the issues related to ontologies supporting agent communication, sensor integration and affective sensing. Experimental results have showed the proper functioning of this system, whose novelty consists in merging together a multi-agent system, an ontology-based mixed environment representation and reasoning engine, and a rule-based model of the user's affective feedback.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: *Multiagent system.*

General Terms
Design, Experimentation, Human Factors, Languages.

Keywords
Smart House, Multi Agent Systems, Context Awareness, Ontology, Affective Computing.

1. INTRODUCTION
The developments in the area of Context Aware Systems stimulated the interest in Ambient Intelligence, particularly in Smart Housing. This is especially beneficial for the nowadays overwhelming amount of information that demands immediate decision making.

In order to increase one's home comfort, some decisions should be automatically taken by intelligent systems. An important drawback of the current available systems is that they are not flexible enough to observe if their predefined behavior is the desired one or not. At that point they may become annoying and the user feels anger or frustration. The social meaning of that reaction is obvious for a human, and even for some of our pets, but nowadays electronic systems are affectively impaired.

The ontological representation of reality is an emerging concept that promises a better understanding of the user and user context and is a good basis for behavioral adaptation. Using ontologies also allows the integration of reasoning engines as part of the decision making process.

Context Aware Systems intrinsically handle a huge amount of information: data acquisition from sensors, status checking for various objects and entities, users' feedback, commands transmission for the actuators, locally stored general and specific ontologies.

In this context, the most appropriate paradigm for designing and implementing such systems is the multi-agent one. By its static and mobile agents and communication protocols, the system can solve the whole task of the intelligent house, ensuring the consistent coordination of all the logic and physical layers.

The developed multi-agent system was tested on the following scenario: Maria is an old and speech impaired person. She is invited to her friend Laura that has a smart home. As Maria is a welcomed guess, the system will authorize her to personalize the system's behavior. One of the Smart House System's predefined rules closes the blinds when the outside light has the same intensity as inside. Maria likes to look outside the window and so, when the first decision of the system to close the blinds is triggered (at sunset, for instance), she will display immediately (in the next minute) a negative emotion (i.e. anger), showing her disapproval. The smart home will adapt to this new behavior.

EATIS'09, June 3–5, 2009, Prague, CZ.

Aware of her negative state, a pet robot will display a funny face to cheer her up.

The paper is organized as follows. In Section 2 a short overview of previous accomplishments in this field is presented. Section 3 offers a detailed presentation of the Smart House architecture, both on hardware and software levels. In the Conclusions section we discuss and expose the lessons learned from the development of this system.

2. RELATED WORK

Jade-Leap was used in the *mySAM* [13] project, to support the deployment of context-aware agents, which can learn to take decisions in a dynamic, changing environment. The stated goal of this project was to develop a multi-agent system, comprising several agent societies, which are able to share their common knowledge about the context, in order to take actions based on the relevant context aspects.

Another project is *C@sa* Agent-Based Home Simulation and Control [4]. It aims at developing a context-aware pervasive system, which is able to control the house behavior according to the context information. The MAS developed for the C@sa, is based on a layered organization of different types of agents: operators, supervisor and interactors.

The CHIL project (Computers in the Human Interaction Loop) [14] represents an implementation of an agent-based system for smart in-door environments. The CHIL architectural framework comprises a layered approach based on three tiers: a sensors tier, a tier of perceptual components used to determine the identity and location of people and objects, a tier of agents that model and keep track of higher level context information.

However these architectures do not have remote monitoring and do not affectively control their behavior.

3. THE SMART HOUSE ARCHITECTURE

3.1 The JADE Platform

JADE differentiates itself from the other evaluated platforms (Agent Development Kit (Tryllian), April Agent Platform, Comtec Agent Platform, FIPA-OS, JACK Intelligent Agents, JAS, ZEUS), by satisfying all the criteria imposed by the evaluators [12]. It is also a very popular platform, providing commitment to FIPA standards, very good security features, integration of several communication protocols, agent mobility support. Moreover, it is a free platform (Open Source, GNU General Public License), very well documented and continuously developed, improved and maintained, not only by the developers from Tilab, but also by JADE community members. According to the evaluation in [10], JADE is the best option since it supports the development of ontologies which can be designed using Protégé and then adapted to JADE using JadeJessProtégé plug-in.

Another evaluation [11], based on two sets of requirements: the compulsory requirements (mobility, FIPA compliance), and the ranking requirements (security, heterogeneity and scalability), showed that Jade-Leap multi-agent platform had the highest rankings, while providing most of the needed characteristics. This is why we found JADE useful for implementing the multi-agent system for the Smart House.

3.2 Smart House Multi Agent Architecture

The multi agent architecture used for developing the Smart House System (Figure 1) shows the layers decomposition and the relationships between them. The layers and the relations between them are established based on Smart House needed functionalities.

3.2.1 Interaction Layer

This layer handles the human-robot interaction, and the human-system interaction. It features high level capabilities for interaction between entities in the house.

This layer provides various facilities: registration of physical entities such as user or device to the system, registration of new tasks or missions (e.g. switch on/off the appliances). It also provides visual representation of the house environment to the user.

3.2.2 Context Information Layer

The Context Information Layer manages context acquisition process, aggregating basic context from various sources. This layer provides the basic context model, whose goal is to represent the context based on data coming from the sensors.

The purpose of this layer is to create a context model, which provides a unique and uniform representation for context information, independently of the particular context source: sensor or device. The Interpreter Agent residing at this layer aggregates the data received from the Sensor layer according to the granularity or scope.

The Information Map Agent manages the context information provided by the context residing in the house environment: objects (doors, walls and windows), appliances, devices, inhabitants (identity, preference, and localization) but also from other context-information providers (e.g. whether prediction service). The information gathered by the Information Map Agent is communicated to the Presentation Layer Agents.

3.2.3 Reasoning & Decision Layer

At this layer, by means of inference rules, high-level context information is inferred from basic sensed contexts and context conflicts are solved. This layer stores the full context model in the Context Knowledge Base while also maintaining the knowledge base consistency.

Therefore, the context Knowledge Base provides updated and persistent knowledge storage by storing the context information of every entity in the system. The Knowledge Base Agent, operating at this layer, holds the responsibility for reasoning tasks. The outcomes of inference procedures are communicated to the Decision Agents, where they are turned into context-specific actions or missions. At this level, user preference is learned using implicit machine learning techniques.

This layer provides the following functionalities: Decision Making, Rule Expert System (performing reasoning over pre-defined rules definition and dynamic rule generation based on available contexts), Reasoning about the state together with task goals and outcomes of possible actions.

The action or mission resulted from the reasoning process carried out by the Brain Agent is communicated to the Action & Mission Layer which records the action/mission, enforcing the appropriate agents to carry out the required tasks.

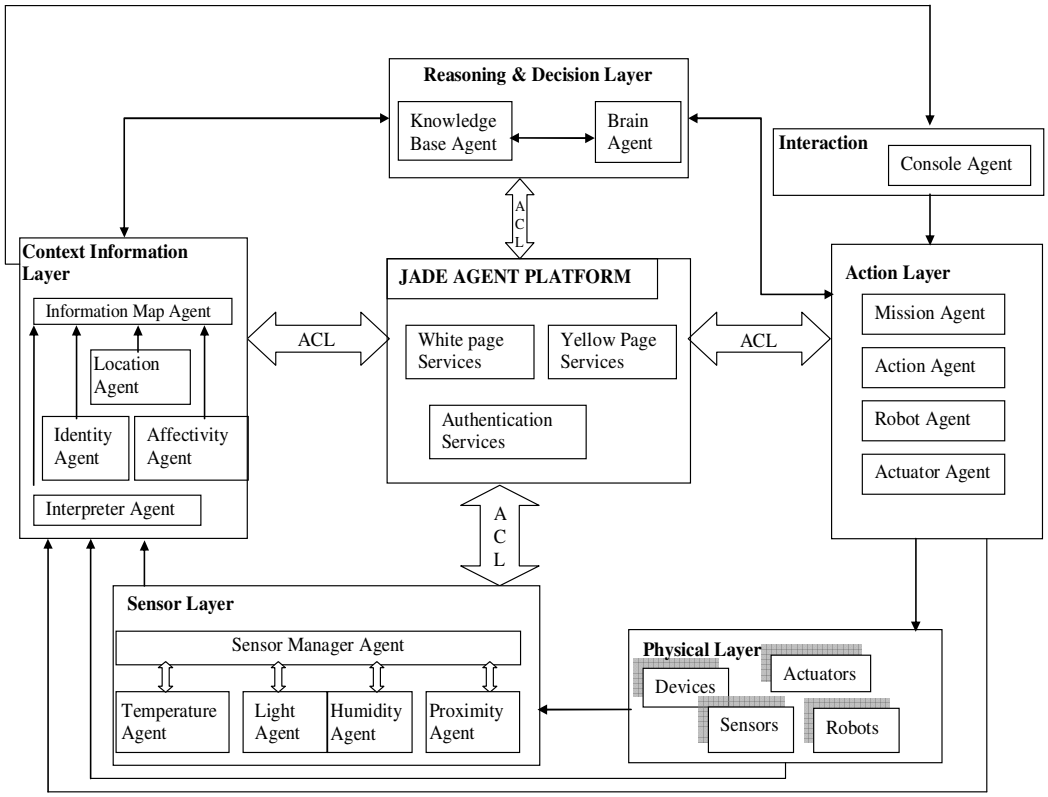

Figure 1. The Smart House Multi Agent Architecture

3.2.4 Action & Mission Layer

This layer deals with the performance of actions or missions upon the environment in order to change the current state.

Its purpose is to present system services (actions or mission) which can be provided by the system. It also enforces the decisions taken at the Reasoning & Decision Layer regarding the actions that should be performed on context change.

3.2.5 Sensor Layer

The Sensor Layer manages the flow of sensor information from the Physical Layer to the Context Information Layer. The Sensor Agents residing at this layer gather data regarding specific sensor information: sensor data, timestamp, sensorID, sensor accuracy (credibility). Sensor malfunctionalities and/or failures have to be detected at this layer.

3.2.6 Agent Platform Layer

It is the deployment platform for all the agents in the Smart House System and provides transparent communication between agents deployed on different devices.

3.2.7 Physical Layer

This layer comprises physical sensors, various objects, devices, appliances and actuators (e.g. air conditioning). Devices are managed by Device Agents who gather device data and transforming the data into the appropriate format, before sending it to the Context Information Layer.

Physical sensors measure the information required by the system and communicate the raw data to the Sensor layer. The physical layer performs the actions communicated by the Action Layer using actuators or device controllers (e.g. for closing the shutters, switch off the light, adjusting the sound volume).

3.3 Smart House Multi-Agent System Design using TAOM4E

TAOM4E stands for Tool for Agent Oriented Visual Modeling for the Eclipse platform. It supports the Tropos agent-oriented software engineering methodology. The development process in Tropos consists in five phases: Early Requirements, Late Requirements, Architectural Design, Detailed Design and Implementation. TAOM4E was developed in order to support a consistent, powerful and customizable software development process. Its features recommend it as a useful support tool for the multi-agent system developed in our Smart House.

In the Late Requirements Analysis phase, the actor diagram from the previous phase (Early Requirements Analysis) is extended by modeling the system in relation to its environment. The decomposition, means-end and contribution analysis goals are performed on the system's goals, as captured in Figure 2. In this phase new system actors are introduced, which are responsible with the core system goals accomplishment: Smart House System, Control System and Monitoring System.

Figure 2. Smart House's goal decomposition using TAOM4E

The Smart House System actor interacts with the Interaction Agent and disseminates tasks for the Control System Actor and for the Monitoring System Actor. The Control System Actor relies on Sensor Manager, Device Manger, Actuator Manager and Robots Manager for the execution of plans and accomplishment of specific goals. The Monitoring System builds the Environment Map providing personalized information to the Family Actors by means of the Interaction Agent (Presentation Agent). The Environment Map information is processed by the Control System, which includes the decisional mechanisms for dynamically adapting to the current context information.

3.4 Knowledge Representation

The use of ontologies in context modeling is important because it is independent from any programming language and supports a formal representation of the context. The ontology allows for knowledge distribution and reuse, logical context reasoning (consistency check, subsumption reasoning, implicit knowledge inference) [16], has expressing power (i.e. OWL has cardinality constraints), hierarchical organization, uses standards for efficient reasoning, abstract programming and interoperability [5]. By using reasoning mechanisms the context can be augmented, enriched and synthesized [3]. Moreover it solves heterogeneity, ambiguity, quality and validity of the context data [9].

In order to design the context ontology we used as starting point the SOCAM [15][6] ontology and we extended it as follows:

- replacing the User class with the Actor class and Group, Individ, Human and NonHuman subclasses;

- adding the State class, with Mental, Physiological and Affective subclasses [2];

- adding NeuralNetwork (for later use) and Sensitivity [13].

We applied the ontology splitting in two main layers described in [11][12]:

- the upper layer – general for context aware applications, and
- the lower layer – dedicated to the Smart House, as can be seen in Figure 3 a) and b).

For the moment Jadex supports only RDF ontology-based message communication. Compared to OWL, RDF is less expressive (indicates only subject-predicate-object relations, without restrictions like existential, universal, cardinality etc. and no characteristics for the properties). In our application, the inter-agent communication ontology is the one depicted in Figure 4. It additionally contains the classes needed for inter-agent communication.

We used Nuggets XML Content Language for developing ontology-based agent communication. This is specific to Jadex. The communication ontology supports access to a structured model of the knowledge domain and facilitates the creation of the JavaBean objects [1]. To export the concepts in a Nuggets XML compatible format, we used Beanynizer, a Protégé plug-in tool.

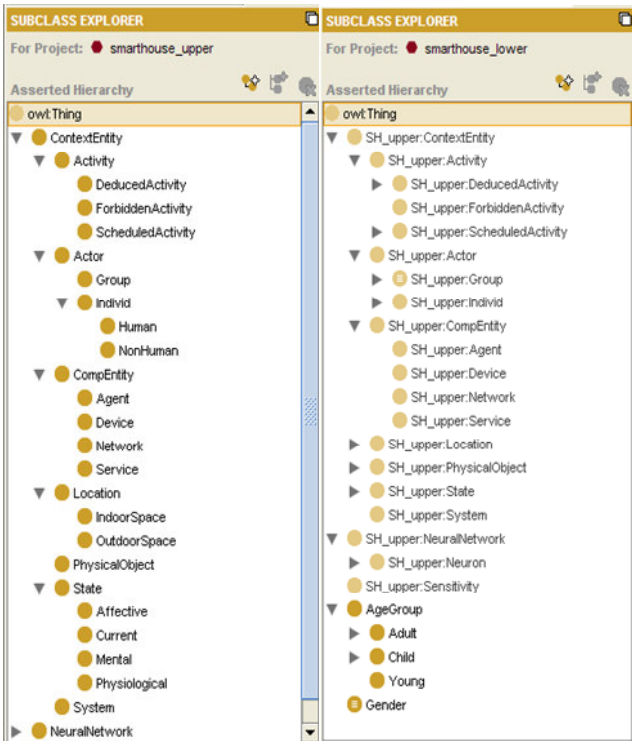

Figure 3. The Smart House context ontology: a) upper (left) and b) lower (right)

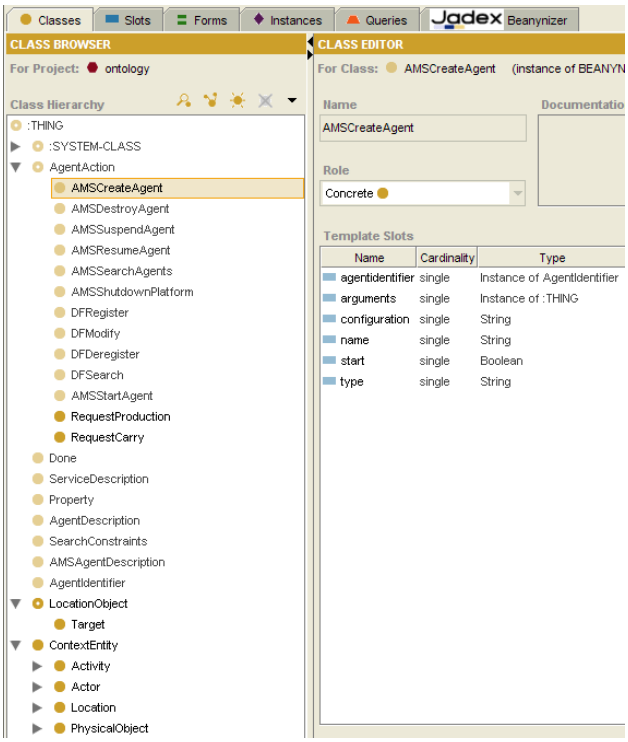

Figure 4. The Smart House agent communication ontology

3.5 Rule and Behavior Expert System

In order to support the reasoning and decisional mechanisms for the Brain Agent residing at the Reasoning & Decision Layer, the Smart House System integrates two subsystems:

- a System that persistently stores rules and context information history;
- an Expert System.

The abstract diagram of these subsystems is shown in Figure 5.

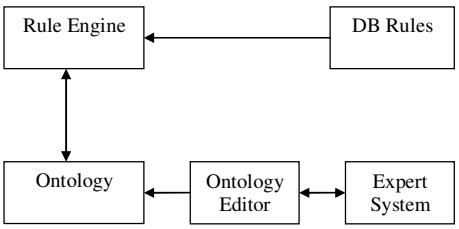

Figure 5. DB and Expert System

The Rule Engine is based on the Jena2 [8] inference subsystem, which integrates a range of inference engines or reasoners (Transitive reasoner, RDFS rule reasoner, OWL, OWL Mini, OWL Micro Reasoners, DAML micro reasoner, Generic rule reasoner). The Rule Engine runs the rules stored in the DB Rules Repository and monitors the changes that occur in the values of the ontological context data. The Rule Engine supports inference on the context objects and then updates the deduced context information back into the ontology model.

The DB Rules component stores the domain rules in a given format (as required by Jena) and includes parameters to be modified by the Expert System (weight matrix values).

The Ontology stores the concepts, represented as classes and their relations. As the ontological model is updated with new values for the context elements (obtained from sensors, devices and other context information sources), the old values are recorded in the persistent ontology database in association with their timestamp.

The Expert System is learning according to the feedback from the user, what would be the best response that user would expect from the system. The neural network is trained by using a set of general predefined examples. After that, the system runs and adapts online based on the user's feedback (i.e. satisfaction level measured from his affective reactions on the systems behavior).

The Ontology Editor reads the characteristics of the neural network (i.e. weight matrix, number of hidden layers, threshold function type) and stores it in the Ontology as the values change in time.

3.6 User Interface Design

The Smart House System includes several user interfaces meant to provide support to the visual representation of the context information provided by sensors, devices and objects and also visual interactivity with the System as a whole (by submitting new plans or missions, or by issuing specific commands for the indoor objects).

Two of the interfaces are designed to work on desktop computers. One is the general sensor-device GUI shown in Figure 6.

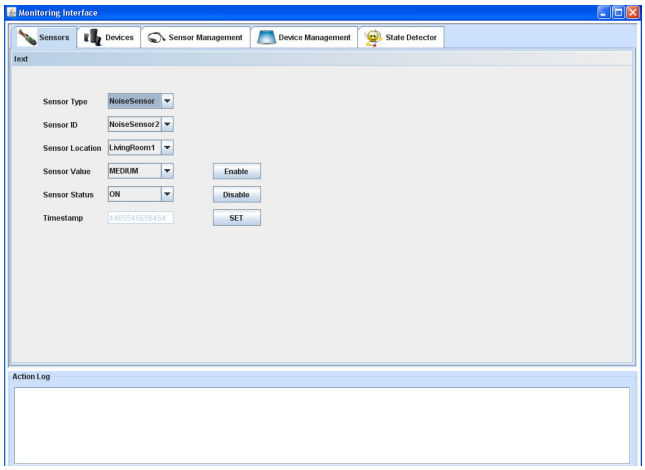

Figure 6. Monitoring GUI (Sensor State - Visualization & Update)

The interfaces are designed to be invoked from Jadex [7] agents. They gather interface data dynamically by collaborating with the appropriate agents as detailed in the system architecture diagram. These interfaces support the following functionalities:

- Visual representation of the inhabitants, robots, objects and devices (Figure 7);
- Visual representation of the house rooms and objects topology;
- Visual representation of deduced context information such as current activity, deduced emotional state;
- Monitoring the indoor objects' and devices' status (Doors, Windows, Appliances, Home entertainment Devices, Sensors);
- Monitoring the House Inhabitants' emotional State;

- Sending Feedback back to the system, by specifying the inhabitants' emotional state;
- Mission Management.

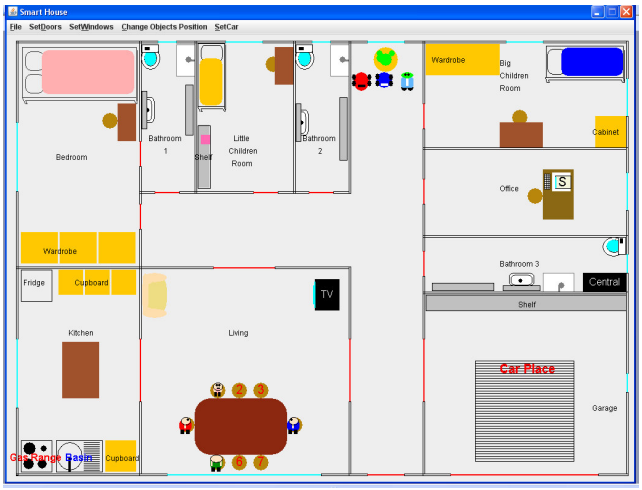

Figure 7. Smart House Simulation

In order to support mobility in the Smart House System, a more simplified interface (see Figure 8) is designed to be run on Mobile Devices. This interface allows the user to monitor and issue commands to the Smart House System, even when the user is remotely located.

Figure 8. Mobile Agents Interface on Sun Java Wireless Toolkit (login – left, sensor information monitoring - right)

Due to the fact that Jadex does not fully support agent mobility, the Mobile Phone Interface Agent was developed in Jade. The capture of the J2ME mobile graphical interface is presented in Figure 8.

3.7 The Jade-Leap Based Monitoring and Control Mobile Application

We developed a Jade-Leap mobile agent system (Lightweight) for monitoring and controlling the smart house sensors and devices. We implemented an authentication mobile agent running on the FrontEnd container and communicating with the BackEnd authentication agent. The mobile monitoring agent requests context information from the ontological agent (deployed on the BackEnd Container) and displays it in a graphical form on the mobile device screen.

4. CONCLUSIONS

When integrating ontologies with multi-agent systems we noticed difficulties as we used different types of ontologies. For the Jadex agents, the inter-agent communication is based on the Nuggets XML codec and the Beanynizer plug-in that automatically converts the ontology concepts in Java Beans. Jadex requires an RDF type ontology, lacking the compatibility with OWL (at this moment). So we deployed two ontologies: an OWL ontology for storing the context elements and reasoning and a RDF ontology for inter-agent communication. The synchronization between them is not assured.
Some other difficulties that we observed are:
- the Jadex platform does not fully support BDI mobile agents development;
- the user defined rules in Jena Framework are not expressive enough, i.e. for a large number of context elements the rule number increases exponentially;
- it is difficult to reuse already defined rules when the firing conditions change (for instance, when adding a new context element all the rules must be rewritten).

The aim to overcome these limitations brings about the premises for new improvements of our project. In order to optimize and personalize the Smart House's behavior, we implemented the context change rules as a Neural Network which is a more flexible approach when new context elements are included in the system. Until Jadex will provide full support for developing BDI mobile agents, due to the fact that the underlying agent platform is a JADE platform, we developed the mobile agents in Jade.

In conclusion, we developed an agent based Smart House System using TAOM4E and a Jade-Leap platform for remote monitoring the Smart House. We implemented a central ontology for context representation and for supporting the behavioral changes according to user's affective feedback. We also emphasize the issues related to developing an agent ontology, sensor integration and affective sensing.

The application was tested on the scenario depicted in the Introduction of this paper. We have used a hybrid wireless sensors network with light, temperature, humidity and proximity sensors. The results proved the appropriate functioning of the Smart House system.

5. ACKNOWLEDGMENTS

We thank Mihai Schiţcu for his contribution to the wireless sensor network. This work benefits by the support of the national CNCSIS type A number 1566.

6. REFERENCES

[1] Beanynizer, http://vsis-www.informatik.uni-hamburg.de /projects/jadex/jadex-0.96x/toolguide/ tools.beanynizer.html

[2] Benta, K.I., Rarau, A., Cremene, M. 2007. Ontology Based Affective Context Representation. In Proceedings of EATIS 2007, ISBN 978-1-59593-598-4, 14-17 May, Faro, Portugal.

[3] Buriano, L. Marchetti, M. Carmagnola, F., Cena, F., Gena, C., Torre, I. 2006. The Role of Ontologies in Context-Aware Recommender Systems. In: IEEE International Conference on Mobile Data Management, pp. 80.

[4] De Carolis, B., Cozzolongo, G., Pizzutilo, S., and Plantamura, V.L. 2005. Agent-Based Home Simulation and Control. Springer, LNCS, 2005, vol. 3488, pp. 404-412.

[5] Ejigu, D., Scuturici, M., Brunie, L. 2007. An Ontology-Based Approach to Context Modeling and Reasoning in Pervasive Computing. In Proceedings of the Fifth IEEE international Conference on Pervasive Computing and Communications Workshops, Washington.

[6] Gu, T., Pung, H.K., Zhang, D. 2005. A Service-Oriented Middleware for Building Context-Aware Services. Elsevier JNCA, vol. 28, issue 1, pp. 1-18.

[7] Jadex, jadex.informatik.uni-hamburg.de

[8] Jena, http://jena.sourceforge.net/downloads.html

[9] Krummenacher, R., Strang, T. 2007. Ontology-Based Context Modeling. In Proceedings of the 3rd Workshop on Context Awareness for Proactive Systems, Guildford.

[10] Leszczyna, R. 2004. Evaluation of Agent Platforms. Technical Report, European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizen, June 2004, Ispra, Italy.

[11] Liljedahl, A. 2004. Evaluation of Multi-Agent Platforms for Ubiquitous Computing. Master Thesis, Thesis no: MCS-2004-12, June 2004, Blekinge Institute of Technology, Sweden.

[12] Nguyen, G., Dang, T.T, Hluchy, L., Laclavik, M., Balogh, Z., Budinska, I. 2002. Agent platform evaluation and comparison, Technical Report for Pellucid 5FP IST-2001-34519. Jun 2002, Bratislava, Slovakia.

[13] Oana, B., Philippe, B., Olivier, B. 2005. Representing Context in an Agent Architecture for Context-Based Decision Making. In Proceedings of CRR'05 Workshop on Context Representation and Reasoning, July 5, Paris, France.

[14] Soldatos, J.K. 2006. Software Agents in Ubiquitous Computing: Benefits and the CHIL Case Study. In Proceedings of First International Workshop on Software Agents in Information Systems and Industrial Applications (SAISIA), February, Karlsruhe, Germany.

[15] Wang, X.H., Zhang, D., Gu, T., Pung, H. K. 2004. Ontology Based Context Modeling and Reasoning using OWL. In Proceedings of Workshop on Context Modeling and Reasoning, Orlando.

[16] Yau, S. S., Liu, J. 2006. Hierarchical Situation Modeling and Reasoning for Pervasive Computing. In Proceedings of The Fourth IEEE Workshop on Software Technologies For Future Embedded and Ubiquitous Systems, and the Second international Workshop on Collaborative Computing, integration, and Assurance, Washington.