An Adaptive Multi-Agent System Architecture for the Smart Home

Serban Radu

***Abstract*—**A multi-agent model is developed for supporting the architecture of a smart home. The model focuses on embedded devices, sensors framework, actuators and learning algorithms used by smart devices. The agents modeling the smart home receive input from sensors and select a suitable action, which is performed by the effectors. Machine learning techniques make better prediction and also adaptation. The employed smart home model discovers patterns of behavior in the user activities and has the ability to learn these patterns. Also, the used system is able to adapt to the changes in the discovered patterns and to update the model accordingly. The proposed framework has high flexibility, learning capabilities and easy integration with smart objects. The model is tested in a simulated scenario for the smart home, and the agents' capabilities are improved in time using learning algorithms.

***Keywords*—**intelligent devices, machine learning, multi-agent system, smart home.

I. Introduction

THE multi-agent system proposed in this paper simulates the environment of a smart home and is used to manage the interaction of the devices. Major features of the agents refer to the adaptability to the changing context of the environment, the cooperation and the coordination between agents, obtained by exchanging messages. Agents improve in time their capabilities, by learning users' behavior and by adapting to the dynamic changes in the environment.

The devices in the smart home are modeled as agents, which use the gathered information from the environment to reason and to take a good decision. Interconnected devices in the smart homes can be regarded as a part of the future Internet of Things.

Internet changed into a network of interconnected devices, which use different standards, in order to offer advanced communication abilities, information transfer services, applications analysis. A smart environment is obtained by interconnecting the devices [1].

To obtain global connectivity and due to the device heterogeneity in the smart home, protocols for accessing the resources of the devices are necessary. Changes in the configuration of the devices can be expected, so the protocols should be able to manage the new environment of the smart home. These issues refer to interoperability between different standards, data formats, resource types, heterogeneous hardware and software components, database systems, and different users [2].

The Internet of Things considers devices as being autonomous and intelligent objects, which have sensors for perceiving the environment, abilities for storing information and for making reasons, possibilities to cooperate by sharing information and reacting to changes in the environment [3].

The multi-agent system allows the distribution and the flexibility in the configuration of the devices from the smart home, by abstracting heterogeneous subsystems and system resources for cooperation [4]. Agent oriented programming is used for dealing with the advanced features of the agents and for solving the requirements of distributed devices. Modeling the environment using agents allows the encapsulation of control and the interaction between agents [5].

A multi-agent distributed system, based on the biologic area, in a smart home environment, is described in [6]. An experimental smart home [7], is based on multi-task agents, where each agent addresses a specialized part and coordinates with others. An agent framework for smart homes is presented in [8]. The article allocates five agents, which divide the smart home properties into categories: functions, interfaces, preferences, resources and controls.

Smart devices need processing power, in order to offer their services. This processing power must be provided by small and cheap embedded processors. Smart home enables and wireless sensor technology in the form of a homemade processing module [9], are based on a low-cost microcontroller.

The benefits and risks of smart home technologies, from multiple perspectives, are described in [10]. An algorithm that reconfigures the existing plans, called planning Q-learning [11], is integrated in an intelligent environment for elderly persons.

A deep reinforcement learning model for home automation systems is presented in [12]. An energy management system [13] defines autonomously a policy for selling or storing energy surplus in smart homes.

An adaptive smart home system that utilizes machine learning techniques to discover patterns in resident's daily activities and to generate automation policies that mimic these patterns is shown in [14].

This work was supported by the project GEX2017, No. 28/25.09.2017, AU 11.17.15.

S. Radu is with the Computer Science Department, Politehnica University of Bucharest, Romania (phone: 004-021-4029358; fax: 004-021-3181014; e-mail: serban.radu@ cs.pub.ro).

The current paper develops a multi-agent model for the smart home, in which learning techniques are used for improving the agent behavior. The smart home is organized as an environment with heterogeneous devices, which communicate between them, in order to reach agreements.

The paper is organized as follows: Section II develops the architecture of the smart home; Section III presents how the communication in the multi-agent environment is performed, and Section IV shows the learning techniques used by agents. Section V develops certain scenarios in the smart home, while conclusions and future work are described in Section VI.

II. System Architecture

The architecture is composed from three layers, as described in Fig. 1: the device layer, which deals with the interaction and the control of the smart home devices, the management layer, which contains the data storage component, the application layer, which shows the services offered by the devices to the users.

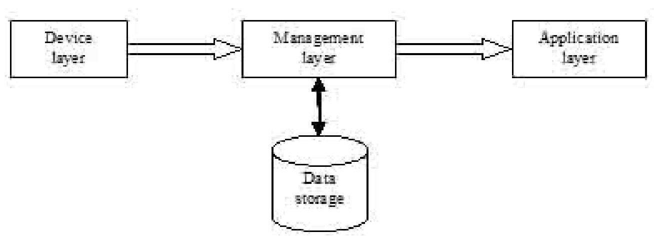

Fig. 1 smart home architecture

The device layer contains simulated or real devices, as well as the environment. This layer contains features of devices and of the environment. Features of the devices refer to the device description and the services offered by the device. Some environmental conditions have to be monitored by the devices. The sensors and the actuators contain the equipment necessary to record data or to make changes in the environment. Each device is managed by a thread. When a new device appears in the system, a new thread is created for the device.

The management layer describes the communication between agents and controls the data storage component. Also, it contains the threads of the system and performs periodically tests upon the system, for discovering any problem that can appear.

The application layer implements the multi-agent system for providing services, for instance the data transmission from different sensors on the Internet.

The agents try to predict the next action of the user, such that repetitive actions of the users are executed automatically. The prediction of the agents is based on previous interactions of the user with the devices. The number of prediction errors should be as small as possible.

The user of the smart home environment interacts with different devices. These interactions can be regarded as a stream of events, some of them appearing repeatedly. These events can be modeled as a stationary stochastic process, which fulfills the Markov assumption.

A prediction problem has the goal to predict what will happen in the future. The input for the predictor is represented by sensor values. Prediction problems can be classification or regression problems. An example of classification problems is event prediction, where the goal is to predict the most probable event, while latency prediction is a regression problem, in which the output takes continuous values.

A pattern discovery approach extracts common patterns from user behavior and learns a predictive model of the next user actions.

The inclusion of the estimation result into the set of possible states allows the user prediction component to guide the learning algorithm.

Reinforcement learning approach gives the optimal choice model of the user, for providing personalized suggestions.

In order to learn patterns of behavior, the agent explores the effects of its actions in time and uses the gathered experience to create policies for optimizing the expected future reward.

Learning a model of user's activities gives the possibility to discover the activities in the smart home. In time, it is possible that the users will change their activity patterns, depending on different factors. It was necessary to adapt to the changes that appear in time and to include user guidance for certain activities.

III. Communication in the Multi-Agent based Platform

The smart home environment needs to exchange messages between its connected sensors and devices. Also, communication is necessary with outside systems, such that a network of smart environments is created. Such a network uses a centralized or a distributed topology. This refers to the manner in which the devices are connected and organized. The functional part deals with the division of the functions between the network devices and describes how the resources are distributed.

The functions of the sensors and the devices are grouped into nodes. In the smart home, sensor node receives information from a sensor and transmits it to a central device. An intelligent sensor processes data before transmitting the information. An actuator node receives information from a central device and controls an actuator, based on this information. A routing node offers routing abilities, by receiving a message and passing it to other nodes. All sensors and smart devices are connected to the routing node. The role of the routing node is to communicate among sensors, actuators and the devices.

Information distribution is necessary to enhance the services offered to the user. When sensors detect certain actions, different events are recorded in the system data storage. A command is sent to the actuators, which offer services to the users. Different scenarios in the smart home determined information to be distributed in the environment.

Routing messages in smart environments implies complex mechanisms. It was necessary to put together different devices in a single network. More than this, the smart router takes decisions, based on the resources of the devices and their processing power.

A distributed agent based system is able to perceive the current context of the devices, having the possibility to exchange messages with other agents and with users. Also, it was necessary to integrate heterogeneous devices and to offer processing power.

The agents are designed according to the BDI model, and have a set of goals, selected from the set of desires. In order to achieve their goals, agents develop plans, as a sequence of actions to be performed, or they provide services or ask for services from other agents.

For a given user goal and a set of beliefs, a set of feasible desires and intentions are generated, in order to select the appropriate agent behavior.

Each agent has different reasoning capabilities, depending on its type. In the proposed system, there are three types of agents: sensing agents, decision agents, and action agents. The agents are constructing efficient plans for goal fulfillment. An agent should have details on others' identity and abilities. It does not need, along its activity, to keep information on all the agents, but only on those it is interacting.

For instance, the beliefs refer to time, location, number of humans, and energy conditions. The desire of a sensing agent, having the task to control the lighting system of the house, is to set up correct illumination conditions for each human activity in the house, with the goal of reducing the power consumption. The intention of the sensing agent can be split in: a) use the thermal sensor to identify the human activity, b) use the light sensor to detect the current illumination level, c) adjust illumination conditions to a proper level for a certain activity, d) choose an energy efficient behavior.

Smart homes are enriched with context-aware services, using artificial intelligence techniques. The devices learn activities from users' behaviors. When these actions are learned, the system recognizes the context and makes suggestions to the user or executes actions autonomously.

Each activity has an associated probability to happen, showing if the activity is frequent or periodic. If the probability of a certain activity is low, that activity is eliminated from the model.

Each activity contains a description of the included events, the constraints between episodes, the temporal relationship between these events, and their duration.

IV. Learning Algorithms Used by Agents

In order to optimize the comfort of the users in the smart home environment, agents employ the reinforcement learning algorithm. The agents revise their strategies, based on observed success or failure. They learn autonomously from rewards, such that the users control upon the system is reduced. For learning a strategy, the agents explore the results of their actions in time and use the gathered experience to create a policy for optimizing the expected future reward.

Because the smart home environment is dynamic, new devices appear and disappear, so the agent is able to learn the model of the environment.

The set of states includes the state of the devices in the current setting. Any change in the environment has as result the transition into a new state. This transition appears either because an event happens or because the user interacts with the system.

Reinforcement learning is applied in the case when each action is associated with an event from the environment. When a state change appears in one of the devices of the smart home environment, the agent takes a decision. The optimal policy is learned on the set of possible states of the devices and the possible actions represent the predictions of the next events.

The reward function is defined considering the user preferences, the energy consumption of the device and the quality of the next state of the system.

Using reinforcement learning, a positive reward is received for each action that is correctly predicted and a negative reward for each incorrect action.

In Q-learning, a reward function provides feedback on actions taken, in order to estimate a ranking of state-action pairs. The Q-learning algorithm converges to the optimal combinations of state-action pairs, after each action is tried enough times. The ranking is due to rewards by matching actions to certain states.

When the Q-values associated with each state-action pair are updated, this means rewarding the actions which give good results. The Q-learning formula used for updating is:

$$Q(s,a) = Q(s,a) + \alpha * [r + \gamma * \max_a Q(s',a') - Q(s,a)] \quad (1)$$

where α is the learning rate, r is the reward obtained by executing action a in state s, γ is the discount factor, $max_a Q(s',a')$ is the maximum Q value for the actions in the next state.

Because the environment is dynamic, the same action may not lead to a desired output, when applied in the identical state. The action selected to be performed is the one with the greatest Q-value. In order to obtain a high reward, the agent prefers actions that were considered good in the past, but, in order to find them, it must try actions that were never selected before.

This leads to the tradeoff between exploitation and exploration. In order to fulfill this tradeoff, two policies are possible: a) the ε-greedy approach, which selects uniformly a non-greedy action, with a probability ε, b) the softmax policy, which uses a given degree of exploration T, for choosing between non-greedy actions, while considering their ranking.

It is better that the agent does not prefer the first action leading to a next state, such that the agent increases the utility obtained in next states. Before the agent tries enough actions, it has an incomplete knowledge of the environment. The agent knows what action to perform, for obtaining a next state, but not what the best actions are. In order to put the agent to try all the actions available before preferring the best ones, it is used a reinforcement learning technique called optimistic initial values. It means that all the Q-values associated with the actions are initialized to a value greater than the expected reward. This measure increases the initial action exploration, because the Q-values are updated to lower values.

In the current approach, the ε-greedy exploration is used, which selects a random action with probability ε and the best action, which is the one with the highest Q-value, with probability 1-ε.

To apply the Q-learning algorithm in a given smart home environment, an appropriate reward structure for showing the users' preferences is defined and a concise representation for the Q-value is found, because the size of the state space highly increases with respect to the devices in the smart home. The actions returned by the prediction algorithm are used to initialize the policy of the decision maker.

Each action executed by the decision maker is assigned a small negative reward. When the user interacts with a device, a big negative reward is assigned. The decision making component learns a policy, which executes automatically the user actions and avoids performing device commands, which are not necessary.

The user can send feedback to the system, in order to adapt the learned activities. The feedback can influence the sequence of learned activities, such that better patterns of behavior are learned in the future.

V. USE CASE SCENARIOS IN THE SMART HOME ENVIRONMENT

To test the smart home platform, different use case scenarios are simulated, in which many devices are placed in the environment. The devices need to describe their services to the platform. Service description is executed using communication primitives, in order to increase the interoperability between the heterogeneous devices and the users.

Scenarios in the smart home involve different sensors, which are distributed to the entrance gate, doors, windows, in the kitchen, at the central heating system. All sensors events are sent to the system data storage. This needs to use the bandwidth of the network, because all the sensors belong to the same network and share the resources of the bandwidth. Location sensors are used to detect a change in the environment.

The smart home environment contains services for the setup, management, and user interaction with the devices. New devices are added in the system and messages are exchanged between devices. By accessing the Internet, the user is capable to remotely control the devices and to program their activities.

Each room of the smart home contains different heterogeneous devices. These devices provide the safety and comfort of the inhabitants, and also optimize the use of resources.

The environment in the smart home implements context-aware services, which can deal with the users' daily activities. The system interacts with the sensors from the environment. Also, the amount of data is huge, and these data should be used by the learning algorithms. Context-aware services are developed in the smart home, using machine learning techniques. The system must be able to learn the activities of the users. After the actions are learned, the system must identify the current situation and act accordingly.

The distributed smart objects from the environment communicate and receive data from sensors. When the user interacts with a sensor belonging to a smart object, for instance the sensors of the lighting system in the house, the behavior of the user is learned in time. The system performs an action, when certain user activity is detected. The system uses these actions, in order to predict more complex user activities. The system learns patterns of activities of the users. For instance, a pattern of activities refer to the order in which a user turns on and off the lights in the house, depending on the path followed by the user in the house, in the morning and in the evening.

An activity learned by the system is to set the home in the evening into the sleep mode. This action is performed by the agents in charge with the lighting system and the heating system. The learning process is performed iteratively and the sequence of actions is gradually adapted, according to the user activities.

Also, in the morning the home is set into the sleep mode, with respect to the lighting system and the heating system. A rule for turning on the sleep mode is if the user will not come back to the house in one hour.

The user is in charge with different scenarios, created from patterns of activities in the smart house. When the user interacts with the smart objects, the associated sensors are activated. Each sensor transmits an event, together with the current time. These events are sent to two components of the system, either to the learning component or to the prediction component. The learning component uses the received events as states for the reinforcement learning algorithm. The prediction occurs after the learning process is finished, for predicting next user actions. The interaction sequence is described in Fig. 2.

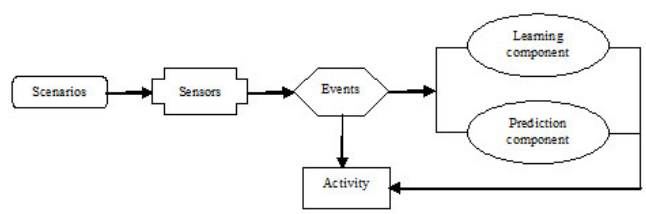

Fig. 2 sequence of processes in the smart home

Another used scenario involves the persons in the smart house having dinner in the kitchen. The sensor detects the presence of the persons and turn on the lights automatically. The sensor associated with the kitchen cabinet, where the plates and the glasses are stored, executes the action of automatically opening the doors of the shelves. Learning takes place with respect to the moment when the doors of the shelves in the kitchen are opened or closed, in order to optimize the sequence of actions in the kitchen.

Another use of the sensors in the smart home refers to the moment when the appliances, like the washing machine and the dish washer, will be turned on, according to the period when the cost of electricity is cheaper. Additionally, this can be remotely controlled, using the smartphone.

A different scenario adapts to the preferences of the inhabitants of the house and uses a database to store the profile of each of the users, including the preferred temperature level in each room. A sensor placed in the room detects the presence of a certain person and adapts the temperature accordingly.

In order to keep the safety of the users, fire detection sensors are placed in the rooms. They detect possible fire, based on data received from smoke and temperature sensors. Additionally, the system generates the current environment state for the specific situation, based on inside and outside temperature, humidity, and other sensor indications.

VI. Conclusions and Future Work

The paper presented a distributed multi-agent system, applied in a smart home environment. The smart devices provide support for the distributed framework. The agents use machine learning techniques to achieve adaptive control. These techniques have an important role in improving the user experience in smart homes. The environments can benefit from machine learning and offer an application domain, in which new techniques can be developed.

Augmented monitoring and sensing capabilities of smart home devices, together with learning techniques, are used to deduce the preferences and habits of the users. This allows smart devices to adapt their operations for increasing the users' quality of life and, in the same time, decrease the energy consumption. Web enabled operation of the smart home enhances the interoperability with the autonomous devices.

Machine learning techniques offer the ability to detect activity patterns, to execute automatically these patterns, and to adapt to possible changes that can appear in time in the users' behavior.

The simulations show that the system is capable to learn and predict the activities of the user, based on the time of the actions and also on sequences of user actions. In the simulations performed, the distributed multi-agent system shows the decrease of the network load and of the energy consumption.

Smart home services improve the life style of the users, allowing to control and to monitor remotely the home devices. For instance, a smart home can automatically close the windows and lower the windows blinds, based on weather forecast.

Future work will be directed towards personalization, privacy and security issues. Personalization and privacy in smart homes is based on user profiles. The security approach will be explored, because smart home devices can be remotely controlled and personal information can be deduced from these devices.

Also, future work will be oriented towards learning models of device performance and energy consumption. The learning algorithm will take into account the situation when different persons will use the smart devices simultaneously.

Acknowledgment

This work was supported by the project GEX2017, No. 28/25.09.2017, AU 11.17.15.

References

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", Technical Report, Cloud Computing and Distributed Systems Laboratory, University of Melbourne, 2012.

[2] A. Katasonov, O. Kayova, O. Khriyenko, S. Nikitin and V.Y. Terziyan, "Smart semantic middleware for the internet of things", International Conference on Informatics in Control, Automation and Robotics, Intelligent Control, Systems and Optimization, pp. 169-178, 2008.

[3] G. Fortino, A. Guerrieri and W. Russo, "Agent-oriented smart objects development", 16th IEEE International Conference on Computer Cooperative Work in Design, pp. 907-912, 2012.

[4] T. Leppanen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Narhi, J. Riekki and T. Ojala, "Mobile agents for integration of internet of things and wireless sensor networks", IEEE International Conference on Systems, Man, and Cybernetics, pp. 14-21, 2013.

[5] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Badica and M. Ivanovic, "Agent-Based Computing in the Internet of Things: a Survey", Studies in Computational Intelligence, 2018.

[6] C. Hannon and L. Brunell, "A Distributed Multi-Agent Framework for Intelligent Environments", Journal on Systemics, Cybernetics and Informatics, vol.3, no.1, 2005.

[7] D. Cook, A. Crandall, B. Thomas and N. Krishnan, "CASAS: A Smart Home in a Box", IEEE Computer, vol. 46, no. 7, pp. 62-69, 2013.

[8] N. Liang, L. Fu and C. Wu, "An integrated, flexible, and Internet-based control architercture for home automation system in the Internet era", Proceedings of IEEE International Conference on Robotics and Automation, ICRA '02, vol. 2, pp. 1101-1106, 2002.

[9] A. Trevennor, "Smart Home Enablers", Practical AVR Microcontrollers 2nd ed. Springer Link, pp. 305-331, 2012.

[10] C. Wilson, T. Hargreaves and R. Hauxwell-Baldwin, "Benefits and risks of smart home technologies", Journal Energy Policy, vol. 103, pp. 72-83, 2017.

[11] M. Hassan and M. Atieh, "Action prediction in Smart Home based on Reinforcement Learning", Lecture Notes in Computer Science, vol. 8456, pp. 207-212, 2015.

[12] V. Gokul, P. Kannan, S. Kumar and S.G. Jacob, "Deep Q-Learning for Home Automation", International Journal of Computer Applications, vol. 152, no. 6, 2016.

[13] H. Berlink and A.H.R. Costa, "Batch Reinforcement Learning for Smart Home Energy Management", Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, pp. 2561-2567, 2015.

[14] P. Rashidi and D.J. Cook, "Keeping the Resident in the Loop: Adapting the Smart Home to the User", IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 39, no. 5, pp. 949-959, 2009.