International Journal of Smart Home
Vol. 9, No. 1 (2015), pp. 217-226
http://dx.doi.org/10.14257/ijsh.2015.9.1.23

Development of a Smart Home Context-aware Application: A Machine Learning based Approach

M. Humayun Kabir, M. Robiul Hoque and Sung-Hyun Yang*

Department of Electronic Engineering, Kwangwoon University
Seoul, Republic of Korea
shyang@kw.ac.kr

Abstract

Context-awareness is an important characteristic of smart home. Several methods are used in context-aware application to provide services. The main target of smart home is to predict the demand of home users and proactively provide the proper services by computing user's context information. In this paper, we present a context-aware application which can provide service according to predefined choice of user. It uses Mahalanobis distance based k nearest neighbors classifier technique for inference of predefined service. We combine the features of supervised and unsupervised machine learning in the proposed application. This application can also adapt itself when the choice of user is changed by using Q-learning reinforcement learning algorithm.

Keywords: *Context-awareness, Reinforcement learning, Smart Home.*

1. Introduction

Researchers in different fields make a lot of contribution for context-aware system. Context-aware systems are able to adapt their operations to the current context without explicit user intervention and thus aim at increasing usability and effectiveness by taking environmental context into account [1]. The goal of the context-aware system in smart home is to provide services that maximize the user's comfort and safety while minimizing the user's explicit interaction with the environment as well as the cost of the service. In general, a set of rules have to be specified for the possible contextual configurations and assign each rule to a specific service in context-aware system. This is a common problem to determine and limit the set of possible context configurations. Instead of rule base approach we use machine learning to predict all possible and meaningful context configurations. This machine learning approach can use previous choice about service and can adapt itself by new choice about service from user feedback information. ACHE (Adaptive Control of Home Environments) is an adaptive house that controls the comfort systems of a home such as lightning, ventilation and air and water heating [2,3]. The objectives of ACHE are the prediction of inhabitant actions and the decrease of energy consumption. It tries to decrease the need for manual control of the systems by anticipating the need to adjust them. The predictors use feed-forward neural network and, in some cases, also look-up tables in combination to make predictions. The MavHome (Managing an Intelligent Versatile Home) project uses multi-agent systems (MAS) and machine learning techniques to create a home environment that is able to act as a rational agent [4,5]. The operation of MavHome is divided in three separate phases: knowledge discovery and initial learning, Operation and Adaptation and continued

* Corresponding Author

ISSN: 1975-4094 IJSH

learning. The Essex intelligent Dormitory (iDorm) is a test-bed for ambient intelligence and ubiquitous computing experiments [6,7,8]. The iDorm embedded agent contains the most intelligence in the dormitory. It receives the sensor values, computes appropriate actions using the learnt behavior of the user as a reference and sends the actions through the network to the actuators. The learning is based on negative reinforcement and occurs whenever the user expresses dissatisfaction by changing the actions that the embedded agent has carried out [9]. Reinisch et al. proposed a concept called ThinkHome to apply artificial intelligence to smart homes with the aim of reducing energy consumption [10]. It includes a knowledge base (KB) that stores data about the environment in ontology form and a multi-agent system (MAS) that contains specific agents for different tasks. ThinkHome is still in the conceptual phase and there are no actual implementations yet. Our aim is to use both supervised and unsupervised learning with ontology based context modeling in the development of context-aware application. This application can provide service proactively by using different types of learning in real smart home environment.

The rest of the paper is organized as follows. In section 2, we discuss the details of our proposed method. Section 3 shows the implementation environment followed by an overall conclusion in section 4.

2. Proposed Method

The context-aware system infrastructure generally consists of sensor, middleware and application layers. Each layer is structurally separated and run independently. The sensor layer obtains primitive information from environment monitoring using various sensors. The middleware stores and analyzes primitive information and make a decision to current contexts. Context-aware system uses middleware for context modeling and context reasoning in order to make high level context from raw data. For our context-aware application we use ontology based context-aware middleware which is developed by our lab project. This middleware has several important functional blocks such as resource manager, context provider, command processor, resource repository. Resource manager is responsible to control all sensors, context clients (PDAs, smart phone) and home appliances. Resource manager checks sensor status, store sensor's raw data into database, Resource manager has emergency control module which is used to control emergency situation like fire, high value of sensors (high CO2). Emergency control module creates alert message and sends it to responsible users and agencies. Context provider collects raw data from resource manager and use ontology to make high-level context. Command processor provides an interface between middleware to Application layer. Command processor receives service command from context-aware application layer and analyzes the command. After analysis it sets control parameter related to the context client and home appliances and sends this command to resource manager for execution. Resource repository stores different types of related information of smart home, sensors data, service activity log and user personal profile. In this paper we will focus on context-aware application section. Details description of context-aware middleware is out of scope of this paper. The context-aware application consists of several modules for specific task as shown in fig.1. The application layer provides users with related data and services such as Sleeping service, Morning call service, Grooming service, Dinning service, Entertainment service, Study service and Guarding service according to current contexts. Fig.2 shows context-aware adaptive service block diagram for smart home.

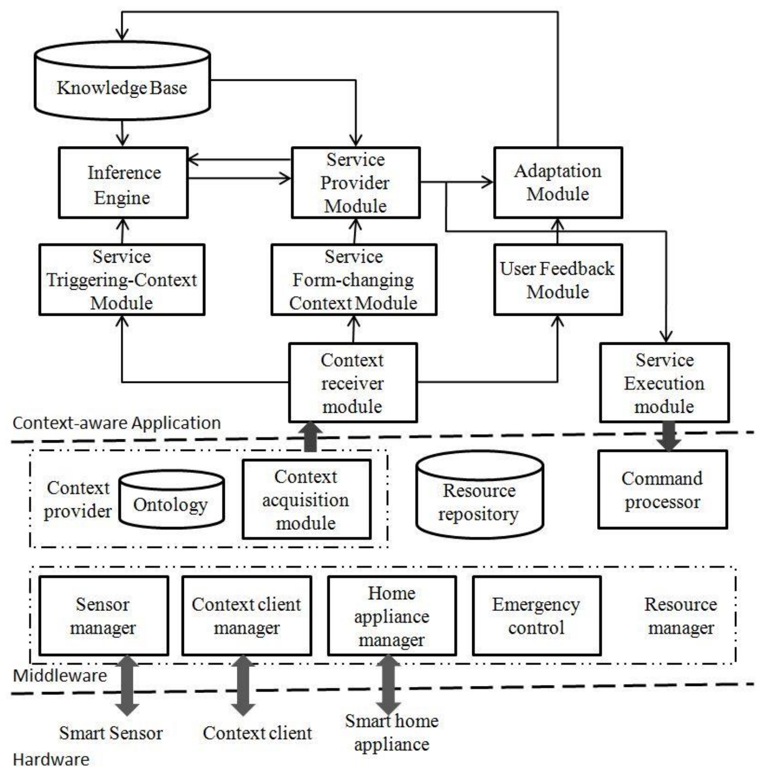

Figure 1. Block Diagram of Context-aware System.

2.1 Context Receiver Module: Context receiver module receives predefined context from context provider in our previously designed middleware architecture. These context information are classified into service-triggering information, service form-changing information and user feedback. Some context information (Person ID, Location, Time, Day and Activity) those are used for selecting services are called service-triggering information [11]. Service-triggering information selects a service in a default form. After selecting the service, the form of this service can be changed depend on user choice/preferences. We can get user choice from user personal profile data. There are several forms of each service. The form of the service needs to be change depend on the choice of the user. So, we need some context information that describes the user choice/preferences. We called these context information as service form-changing context. In our application we use personal profile as one of the service form-changing information. Resource repository section in the context-aware middleware stores the user personal profile data. After providing the service by the application, user's satisfaction can be explicitly or implicitly obtained by user feedback information. A details explanation of user feedback is added in adaptation module and user feedback section.

2.2 Service-triggering Context Module: This module detects service-triggering information element and sends it to inference engine. We have chosen five information elements Person ID, Location, Time, Day and Activity as service-triggering information. These triggering information element values are used to select the service among the seven available services. Each service-triggering information element is specified by a set of values as shown in table I.

Table 1. Service-triggering Information Elements with Values

Context information elements	Context information elements values
Person ID	Father, Mother, Son, Daughter
Time	Morning, Noon, Afternoon, Evening, Night, Latenight
Day	Workday, Holiday
Location	Outside, Bedroom, Living-room, Kitchen, Toilet
Activity	Standing, Sitting, Walking, Reading, Lying, Eating, Watching

We can get N vectors of different sizes depending on the number of service-triggering information elements. Let $E_1=\{x_1,x_2,\ldots\ldots x_{last1}\}$, $E_2=\{x_1\ ,x_2\ ,\ldots\ldots x_{last2}\},\ldots\ldots\ldots,\ E_N=\{x_1,x_2,\ldots\ldots x_{lastN}\}$ are different values of N information elements. We can define each context configuration (C_1, C_2, C_3 ……..C_N) by a set of 5-tuples of different values of information elements such as C_1 {Father, Morning, Workday, Bedroom, Lying}, C_2 {Son, Evening, Workday, Living-room, Reading}, C_3 {Mother, Night, Holiday, Kitchen, Sitting, etc. These context configuration select different services such as C_1{Father, Morning, Workday, Bedroom, Lying} >> MorningCall, C_2{Son, Evening, Workday, Living-room, Reading} >>Study, C_3{Mother, Night, Holiday, Kitchen, Sitting} >> Dinning.

2.3 Service Form-changing Context Module: This module collects service form-changing information element with values shown in table II. Service form-changing context changes the form of services. For example, during morning call if any user does not like bright light then the light pattern will be change according to user's choice. We can also define service form-changing context by a set of 11-tuples as described in service-triggering context module section.

Table 2. Service Form-changing Information with Values.

Service form-changing context elements	Cont. elements values
Person profile: Light	Dim, Off
Person profile: Blind	On, Off
Person profile: TempSummer	Low, Medium, High
Person profile: TempWinter	Low, Medium, High
Person profile: Humidity	Low, Medium, High
Person profile: Ventilation	Low, Medium, High
Person profile: Alarm	On, Off
Season	Summer, Winter
TempSensor	Low, Medium, High

LightSensor	Low, Medium, High
HumiditySensor	Low, Medium, High
CO_2Sensor	Low, Medium, High

2.4 Learning and Mapping: We need to map each service-triggering context configuration to the appropriate service depending on the values of context elements. This context configuration will serve as a training data to k-nearest neighbors classifier. For training sets we consider 28 sets of context configuration of service-triggering information and label them with respect to the choice services.

2.5 Inference Engine: Whenever a new service triggering-context is available then it will send to Inference engine. Inference engine will compare the new context with the train contexts and find the best solution for that context. The best matching context label will be treated as service identity. Inference engine works based on k-nearest neighbors (k-NN) classifier. For distance calculation Mahalanobis distance metric is used.

The k-nearest neighbors (kNN) rule classifies each unlabeled example by the majority label of its k-nearest neighbors in the training set. Despite its simplicity, the kNN rule often yields competitive results in certain domains, when cleverly combines with prior knowledge, it has significantly advanced the sate-of-the-art. The kNN compute simple Euclidean distances assuming the examples are represented as vector inputs in most implementations. Unfortunately, Euclidean distances ignore any statistical regularity that might be estimated from a large training set of labeled examples. A number of researchers have demonstrated that kNN classification can be greatly improved by learning an appropriate distance metric from labeled examples. The Euclidean distance is blind to correlated variables on the other hand the Mahalanobis distance takes into account the covariance among the variables in calculating distances [12]. We will use Mahalanobis distance metric as a replacement of the Euclidean distance for kNN classification. The kNN algorithm is divided into six steps.

Step 1. Value determination of parameter k: The value of k is equal to the number of nearest neighbor to consider. The choice of k depends on the relation between the number of features and the number of cases. A small value of k may influence the result by individual cases, while a large value of k may produce smoother classification outcomes. In our case we considered k=3 for classification of service-triggering context.

Sept 2. Distance calculation between the current context (service-triggering) and all the training samples. Mahalanobis distance provides a way to measure how similar some set of conditions is to known set of conditions. It accounts the covariance among variables. It is calculates as

$$D^2 = (x-m)^T C^{-1} (x-m) \text{--------------(1)}$$

Where, D^2 = Mahalanobis distance

x = vector of data

m = Vector of mean values of independent variables

C^{-1} = Inverse Covariance matrix of independent variables

T = Indicates vector should be transposed

Step 3. Sorting the distance and determine nearest neighbor based on the k-th minimum distance.

Step 4. Assign the majority class among the nearest neighbors to the new context data.

Step 5. Determine the class using those neighbors.

Step 6. Find out the service label and send it to service provider.

2.6 Service Provider Module: Service provider module is responsible to provide service as selected by the inference engine. Service provider uses knowledge base to get the description of the services. Service provider checks the service form-changing context and determines the form of the service by setting the control parameter of home devices which are related to the offered service. Assume Morningcall service is chosen by the application and person is father. The form of this service is determined depends on father's personal profile and present status of sensor and devices. If we can define father personal profile by a set of values as specified in table II is {Dim, On, Medium, Medium, Medium, Medium, On, Summer, Low, Low, Medium, Medium}. With this context, the service form can be determined as {Light (dim), Blind (on), Alarm (on), Aircon (medium), Humidifier (medium), Ventilation (medium)}

2.7 Service Execution Module: Service execution module receives information about services from service provider. Service execution module sends the information to command processor module in our designed middleware architecture for execution.

2.8 Adaption Module: One challenge for the learning algorithm is that the user may change his behavior over time. Consider a resident that used to turn on the coffee maker every day at 7:30a.m. but later changes his habit and turns it on at 6:30a.m. So, it is not always practical to learn the user's behavior off-line instead an adaptive learning algorithm is needed [13]. We apply reinforcement learning (RL) as an unsupervised algorithm which does not need predefined models of the environment and learns the model from user's feedback to the service provided by the system [14]. If the user does not satisfied with the service made by the system, (s)he is more likely to return to the previous service or to switch to a completely new one. There are two types of feedback, positive feedback and negative feedback. This feedback can be received by explicit input from the user or may be estimated implicitly.

Adaptive module receives explicit or implicit user's feedback with every change in the service. Adaptive module records relative rate of negative feedback with service taken in every state for different state-action pairs and is updated according to Q-learning algorithm [15,16].

$$Q(a,s) = Q(a,s) + \propto \left(R(s) + \gamma . \max_{a'} Q(a',s') - Q(a,s)\right) \text{----------}(2)$$

Where, Q(a,s) represents the old value of state-action pairs

α is the learning rate

γ is a discount factor that takes in account of all future rewards

R is the immediate rewards

If Q(a,s) reaches a certain threshold values then the corresponding action(service) and context is added as a train data in knowledge base. Inference Engine will use this data for selecting the service next time.

2.9 User Feedback Module: This module receives context from context receiver module and select explicit user feedback context. In our system user can explicitly appreciate a decision by inputting 0 and demoralize it by -1. If the user does not respond then the system will implicitly calculates the feedback with the following equation

$$Feedback=\frac{\text{The time user stays with the service}}{\text{Total time of the service}}-1 \qquad \text{----------------(3)}$$

User feedback module sends feedback information to adaptation module.

2.10 Knowledge Base: Knowledge base stores trained context configurations set with service label. It also holds the service description. It is updated with new trained context configuration with service, labeled by the adaption module.

3. The Implementation Environment

Our test-bed is constructed in a 93.73 square meter apartment with 2 bed rooms, 1 living room, 1 toilet, 1 kitchen, 1 storage room and hallway. All rooms are equipped with sensors (temperature, humidity, CO_2, illuminance, pressure sensor, and camera) smart device (door lock, blind, smart phone) and smart appliances (TV, refrigerator, washing machine, air conditioner, microwave oven, room heater, etc.). All sensors, smart devices and smart appliances are connected to a residential gateway. The role of the home gateway is to communicate between sensors, smart device, appliances and the control system. Protégé Ontology editor can be used for ontology design, Pallet reasoner for ontology reasoning ,OWL API and Java platform for implementation within this test-bed [17].

Figure 2. Block Diagram of our Testbed

4. Conclusion

Smart home environments are complex and dynamic. It is difficult to design algorithms that are able to control the environment in such a way that user comfort is maximized in every situation. A control system based on machine learning is suitable to achieve adaptive control in smart home. Rule base approach fully depends on rule for making any decision. If current problem not satisfied the rule then it does not provide any decision. We have applied learning mechanism to better mapping each context configuration to the appropriate service. Machine based approach uses previous knowledge for making the decision of the current problem and can add new knowledge for further use.

Acknowledgements

This work was supported by the Industrial Strategic Technology Development Program (10041788, Development of Smart Home Service based on Advanced Context-Awareness) funded by the Ministry of Trade, Industry & Energy (MI, Korea) and the Research Grant of Kwangwoon University in 2014.

References

[1] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems”, Int. J. Ad Hoc and Ubiquitous Computing, vol. 2, no. 4, **(2007)**, pp. 263-277.
[2] M. C. Mozer, “The Neural Network House: An Environment that Adapts to its Inhabitants”, In: Proc. AAAI Spring Symposium on Intelligent Environments, **(1998)**.
[3] M. C. Mozer, “Lessons from an Adaptive Home. In: D.J. Cook and S.K. Das (eds.) Smart Environments: Technologies, Protocols, and Applications”, John Wiley, **(2005)** pp. 273–294.
[4] G. M. Youngblood, D. J. Cook, and L. B. Holder, “Managing Adaptive Versatile environments”, Pervasive and Mobile Computing 1, Special issue on PerCom, **(2005)**, pp. 373–403.
[5] D. J. Cook, G. M. Youngblood, and S. K. Das, “A Multi-agent Approach to Controlling a Smart Environment”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, vol. 4008, **(2006)**, pp. 165–182.
[6] P. Remagnino, H. Hagras, N. Monekosso, and S. Velastin, “Ambient Intelligence: A Gentle Introduction”, In: Ambient Intelligence, Springer New York, **(2005)**, pp. 1–14.
[7] H. Hagras, V. Callaghan, M. Colley, and G. Clarke, A. Pounds-Cornish, and H. Duman, “Creating an Ambient-Intelligence Environment Using Embedded Agents”, IEEE Intelligent Systems, vol.19(6), **(2004)**, pp. 12–20.
[8] A. Holmes, H. Duman, and A. Pounds-Cornish, “The iDorm: Gateway to Heterogeneous Networking Environments”, In: International ITEA Workshop on Virtual Home Environments, **(2002)**, pp. 30–37.
[9] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, “An Incremental Adaptive Life Long Learning Approach for Type-2 Fuzzy Embedded Agents in Ambient Intelligent Environments”, IEEE Transactions on Fuzzy Systems vol.15(1), February **(2007)**, pp. 41–55.
[10] C. Reinisch, M. J. Kofler, M.J. and W. Kastner, “ThinkHome: A Smart Home as Digital Ecosystem”, In: Proceedings of 4th IEEE International Conference on Digital Ecosystems and Technologies, **(2010)**, pp. 256–261.
[11] M. Miraoui, C. Tadj, and C. Amar, “ Dynamic Context-aware service adaptation in a pervasive computing system”, Preceding of Third International conference on Mobile Ubiquitous Computing, System, Services and Technologies, October **(2009)**, pp. 77-82.
[12] Q. Kilian , Weinberger, and L. K. Saul, “Distance Metric Learning for Large Margin Nearest Neighbor Classification”, Journal of Machine Learning Research, vol.10, February **(2009)**, pp. 207-244.
[13] A. H. Khalili ,WU Chen and H. Aghajan “Autonomous Learning of User’s Preference of Music and Light Services in Smart Home Applications”, Behavior Monitoring and Interpretation Workshop at German AI Conf., September **(2009)**.
[14] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction”, MIT Press, Cambridge, **(1998)**.
[15] C. J. C. H. Watkins, “Learning from Delayed Rewards,” PhD Dissertation, King's College, Cambridge, May **(1989)**.
[16] C. J. C. H. Watkins and P. Dayan, “Q-Learning”, Machine Learning, vol. 8, May **(1992)**, pp. 279-292.

[17] H. Knublauch, R. Fergerson, N. Noy, and M. Musen, "Protégé OWL plugin: An Open Development Environment for Semantic Web Applications", Third International Semantic Web Conference, **(2004)**.

Authors

M. Humayun Kabir received his B.Sc (Hon's) and M.Sc degree in Applied Physics, Electronics and Communication Engineering from Islamic University, Kushtia, Bangladesh, in 2001 and 2003 respectively. He is an Assistant Professor at Islamic University, Kushtia Bangladesh. Now, he is pursuing his Doctoral degree at Kwangwoon University, Seoul, Republic of Korea. His main research interests are Context-aware systems for the Smart Home, Embedded Systems, M2M, Sensor Networks, Image processing, and Speech processing.

M. Robiul Hoque received B.Sc (Hon's) and M.Sc. degree in Computer Science and Engineering from Islamic University, Kushtia, Bangladesh, in 2003 and 2004 respectively. He is an Assistant Professor at Islamic University, Kushtia, Bangladesh. Now, he is pursuing his Doctoral degree at Kwangwoon University, Republic of Korea. His main research interests include Context-Aware System, Sensor Networks, Image processing, and Speech processing.

Sung-Hyun Yang received his B.S. and M.S. degree in Electrical Engineering from Kwangwoon University, Seoul, Republic of Korea, in 1983 and 1987 respectively. He completed his Ph.D. from Kwangwoon University in 1993. He is a Professor in Electronic Engineering at Kwangwoon University, Seoul, Republic of Korea. He is a Director of the Ubiquitous Home Network Center, Kwangwoon University. He was a Research Scientist at Boston University from 1996 to 1998. He was Chairman of the Home Network Market Activation Section, Korean Association for Smart Home from 2007 to 2008. His main research interests are Digital logic, Embedded Systems, M2M, Next Generation Ubiquitous Home Networks, and Context-aware systems.