Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances

Jonghwa Choi, Dongkyoo Shin and Dongil Shin

Abstract — *Smart homes integrated with sensors, actuators, wireless networks and context-aware middleware will soon become part of out daily life. This paper describes a context-aware middleware providing an automatic home service based on a user's preference inside a smart home. The context-aware middleware include an appliance controller, a context-aware agent and a scalable browser. The appliance controller takes charge of communication between appliances in the context-aware middleware. The context-aware middleware use OSGi(Open Service Gateway Initial) as framework of the home network. The scalable browser recognize the properties of all rendering device, and it figure out their screen size. We use UIML(User Interface Markup Language) as multiple rendering device. The context-aware agent utilizes 6 basic data values for learning and predicting the user's preference for the home appliances: the pulse, the body temperature, the facial expression, the room temperature, the time, and the location. The six data sets construct the context model and are used by the context manager module. The user profile manager maintains history information for home appliances chosen by the user. The user-pattern learning and predicting module is based on a neural network, which predicts the proper home service for the user. The test results show that the pattern of an individual's preference can be effectively evaluated and predicted by adopting the proposed context mode*[1].

Index Terms — Context-aware middleware, OSGi(Open Service Gateway initial), Neural Network, Smart Home

I. INTRODUCTION

SINCE the computing devices are getting cheaper and smaller, we are dealing with ubiquitous computing, as stated by Mark Weiser [1]. The original concept of home intelligence was mostly focused on network connections. Researchers claim that smart homes will bring intelligence to a wide range of functions from energy management, access monitoring, alarms, medical emergency, response systems, appliance controlling, and even interactive games [2].

Appliances installed in a smart home should be able to deliver enhanced or intelligent services within the home. A fundamental role of "Artificial Intelligence" in smart homes is to perform the underlying monitoring, management, and allocation of services and resources that bring together users and information [3].

Moreover context-aware middleware is needed to offer an unobtrusive and appealing environment embedded with pervasive devices that will help its users to achieve their tasks at hand; in other words, technology that interacts closely with its occupants in the most natural ways to the point where such interaction becomes implicit [4].

We propose a context-aware middleware that provide an automatic home service based on a user's preference inside a smart home. We utilize 6 basic data values for learning and predicting the user's preference in the context of the home: the pulse, the body temperature, the facial expression, the room temperature, the time and the location.

Section 2 gives related research work on context-awareness. Section 3 addresses a context-aware middleware framework, and presents how the context is constructed in the middleware. In Section 4, we present implementation and experimental results. We conclude with Section 5.

II. RELATED WORKS

In order to enable natural and meaningful interactions between the context-aware smart home and its occupants, the home has to be aware of its occupants' context, their desires, activities, needs, emotions and situations. Such context will help the home to adopt or customize the interaction with its occupants. By "context", we mean the circumstances or situations in which a computing task takes place. Context of a user is any measurable and relevant information that can affect the behavior of the user.

Meaningful context information has to be derived from raw data acquired by sensors. This context processing aims at building concepts from environmental and human data sensed by sensors. This intelligence processing is also know as context interpretation and should contain two sub-steps: modeling and evaluation [5, 6]. Raw data is modeled to reflect physical entities which could be manipulated and interpreted. Propositions from the modeling module need to be evaluated against a particular context. Evaluation mechanisms often use artificial intelligence techniques.

A context-aware system can be constructed with several basic components. Most of the middleware components gather context information, processes it and derives meaningful actions from it [7]. Ranganathan and Campbell argued that ubiquitous computing environments must provide middleware support for context-awareness. They also proposed a middleware that facilitates the development of context-aware agents. The middleware allows agents to acquire contextual information easily, reason about it using different logics and then adapt themselves to changing contexts.

[1] The authors are with the Department of Computer Science and Engineering, Seoul 143-747, Korea (e-mail: com97@gce.sejong.ac.kr, shindk@sejong.ac.kr, dshin@sejong.ac.kr).

Correspondence author: Dongil Shin

Manuscript received January 12, 2005

Licia Capra proposed the marriage of reflection and metadata as a means for middleware to give applications dynamic access to information about their execution context [8]. Stephen S. Yau developed a reconfigurable context-sensitive middleware for pervasive computing [9]. Reconfigurable context-sensitive middleware facilitates the development and runtime operations of context-sensitive pervasive computing software [9].

III. CONTEXT-AWARE MIDDLEWARE FRAMEWORK

Figure 1 shows the overall architecture of the context-aware middleware.

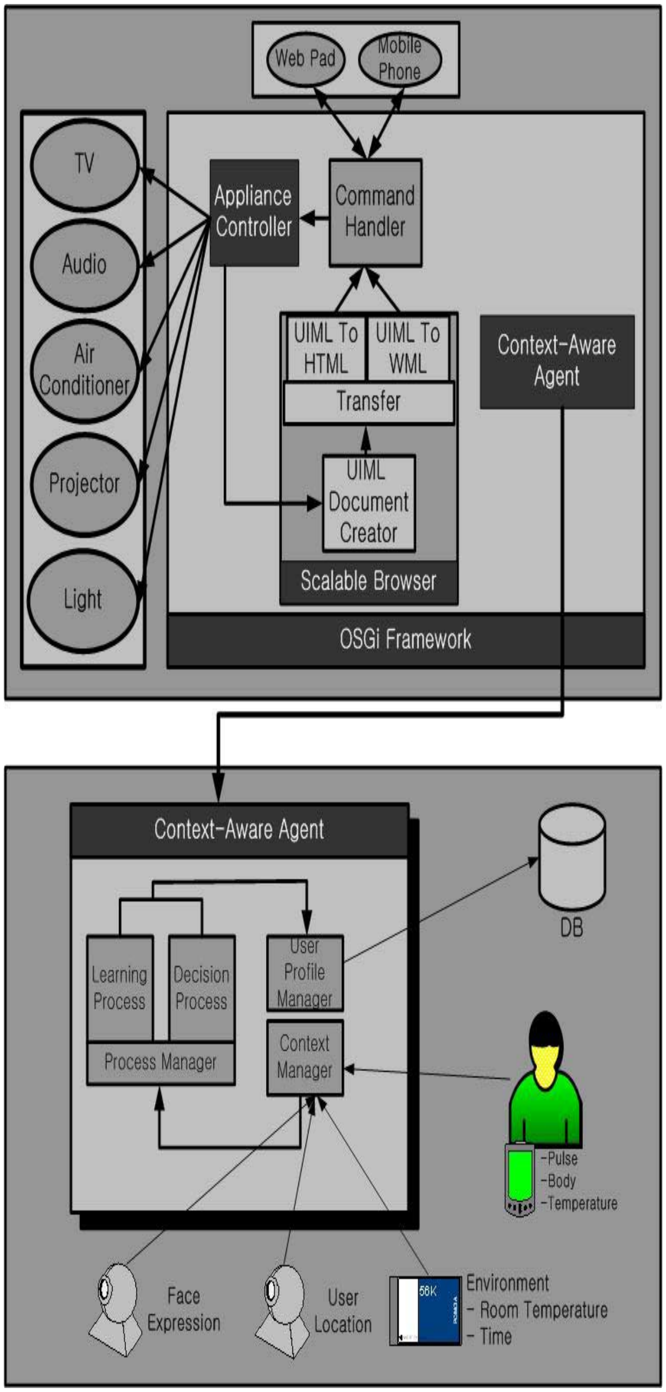


Fig. 1. Overall design of the context-aware middleware

The middleware includes the appliance controller, the context-aware agent and the scalable browser. The appliance controller takes charge of communication between appliances in context-aware middleware. The context-aware middleware used OSGi(Open Service Gateway Initial) as framework of the home network [10]. The scalable browser recognizes the properties of all rendering device, and it figures out their screen size. We use UIML(User Interface Markup Language) as multiple rendering device [11].

A. Appliance Controller and Scalable Browser

The appliance controller takes charge of communication between appliances in the context-aware middleware. The context-aware middleware uses OSGi(Open Service Gateway Initial) as framework of the home network.

The OSGi(Open Services Gateway Initiative) framework supports multiple communication: power line communication, Universal Plug and Play (Upnp), Havi and Jini. The advantage of OSGi is that it contributes to scalable communication of appliances in the smart home [12].

Figure 2 shows the architecture of the OSGi framework.

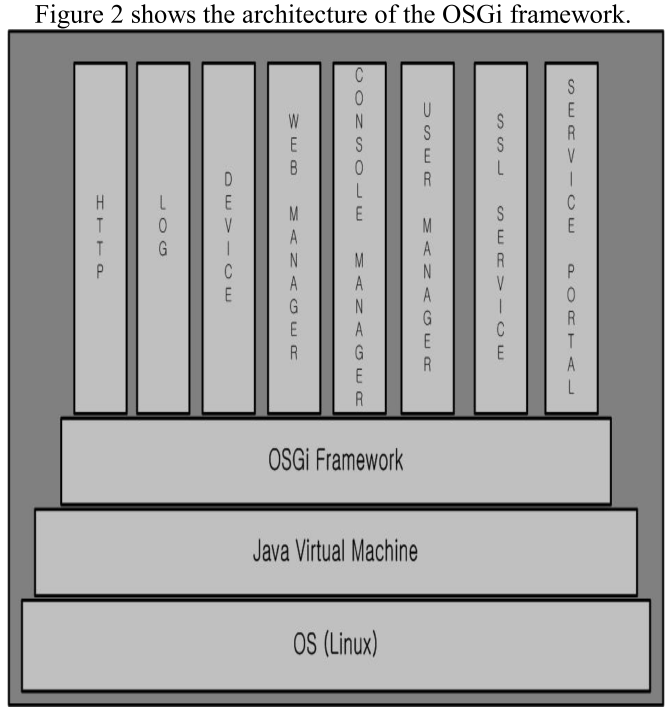


Fig. 2. OSGi Framework

The developer of OSGi has announced the release of OSGi Service Platform Release 3 in March 2003. This specification is designed for Bluetoothe, CAL, CEBus, Convergence, enNET, HAVi, HomePNA, HomePlug, HomeRF, Power Line Commucation and Jini [13]. Also, it supports a home gateway or other device such as a personal computer.

The mission of OSGi is to create open specifications for the delivery of multiple services over wide-area networks to local networks and devices, to accelerate demand for products and services based on these specifications worldwide through the sponsorship of market and user education programs.

Figure 3 presents the bundle structure in context-aware middleware.

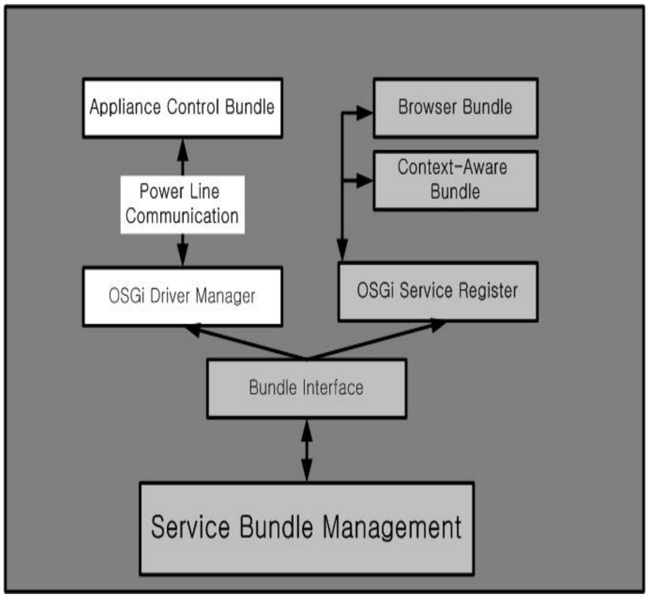


Fig. 3. Bundle diagram in context-aware middleware

The protocol for appliance control is PLC(Power Line Communication), and it runs a PLC driver bundle in the OSGi framework. Our rendering browser use UIML for multiple device rendering. UIML stands for "User Interface Markup Language". UIML is an XML language for defining user interfaces. Figure 4 shows the element structure of UIML.

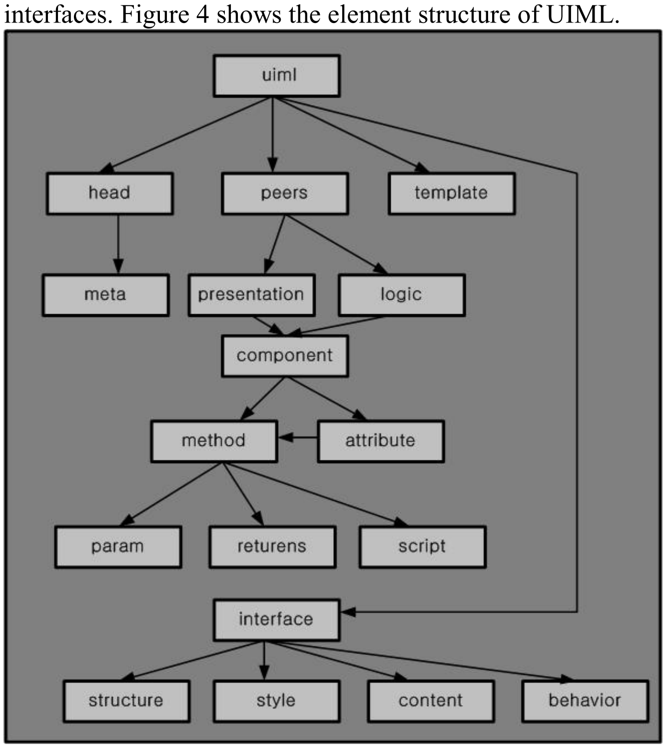


Fig. 4. UIML element structure

Most XML languages are used for defining documents. In other words, they allow programs to break up a lot of words, pictures and other data into useful chunks that can be processed by a program. UIML, on the other hand is used for defining the actual interface elements. This means the buttons, menus, lists and other controls that allow a program to function in a graphical interface like Windows or Motif.

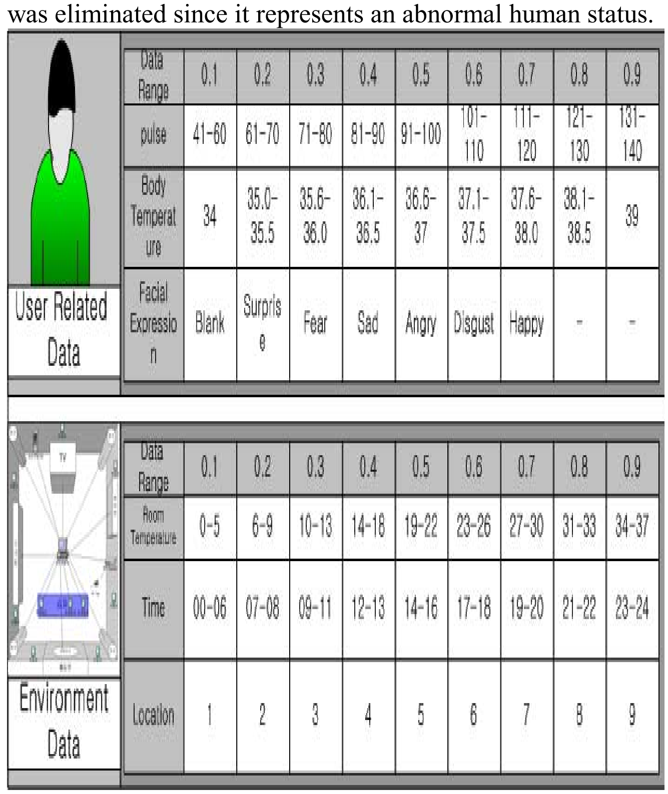
UIML is used to define the location, and design of controls. It also defines actions to be taken when certain events take place. Users create events when they interact with the interface by typing a key on the keyboard or moving and clicking the mouse.

B. Context-Aware Agent

1) Context Definitions

Context's definition is important in context-aware agent. A researcher of context-aware agents proposed a model in which a user's context is described by a set of roles and relations [14]. To attain a user's goal the system must process the user related data along with the environmental data. We proposed a user context of 6 basic data values: the pulse, the body temperature, the facial expression, the room temperature, the time, and the location. The pulse and the body temperature are detected by sensors attached on a PDA (Personal Digital Assistant). The user's facial image is also attained by a small camera from the PDA and transmitted wirelessly. Room temperature is measured by a wall-mounted sensor. We installed 4 cameras to detect the user's location inside a room.

Figure 5 shows the six data sets for the context and they are normalized between 0.1 and 0.9. A pulse below 40 or over 180 was eliminated since it represents an abnormal human status.



Data Range	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
pulse	41-60	61-70	71-80	81-90	91-100	101-110	111-120	121-130	131-140
Body Temperature	34	35.0-35.5	35.6-36.0	36.1-36.5	36.6-37	37.1-37.5	37.6-38.0	38.1-38.5	39
Facial Expression	Blank	Surprise	Fear	Sad	Angry	Disgust	Happy	-	-

Data Range	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Room Temperature	0-5	6-9	10-13	14-18	19-22	23-26	27-30	31-33	34-37
Time	00-06	07-08	09-11	12-13	14-16	17-18	19-20	21-22	23-24
Location	1	2	3	4	5	6	7	8	9

Fig. 5. Data range for the context

A body temperature below 34 or over 40 was eliminated for the same reason. Facial expressions were normalized and categorized as described in [15]. The room temperature was normalized based on the most comfortable temperature which is between 23 and 24 Celsius. The time is normalized based on 24 hours. The location is a user's position in the experimental room.

2) Context Normalization

As shown in Figure 6, the context manager collects the six sets of contextual data, detects garbage data, normalizes the data, and sends the data to the user preference learning and prediction module. If out of range is detected, the context manager automatically recollects the same kind of data.

Context data is delivered in machine-learning algorithm after it passed through the normalization process in the context manager.

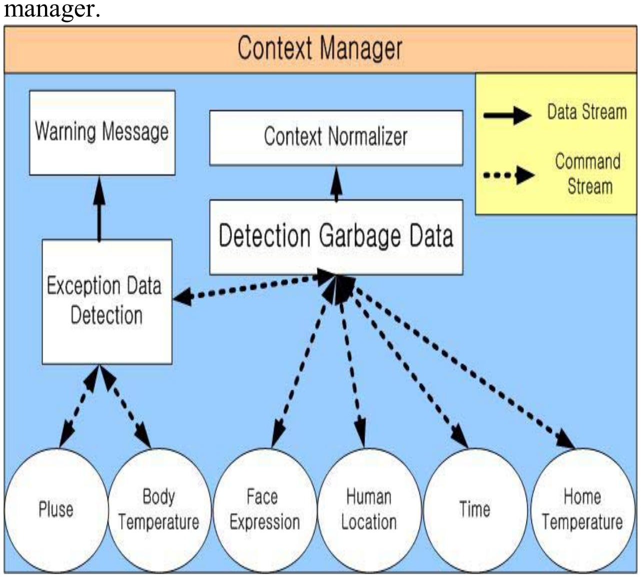


Fig. 6. Context Manager Process

3) Learning and Prediction Module

The user preference learning module uses the context data along with the history of home appliances chosen by the user.

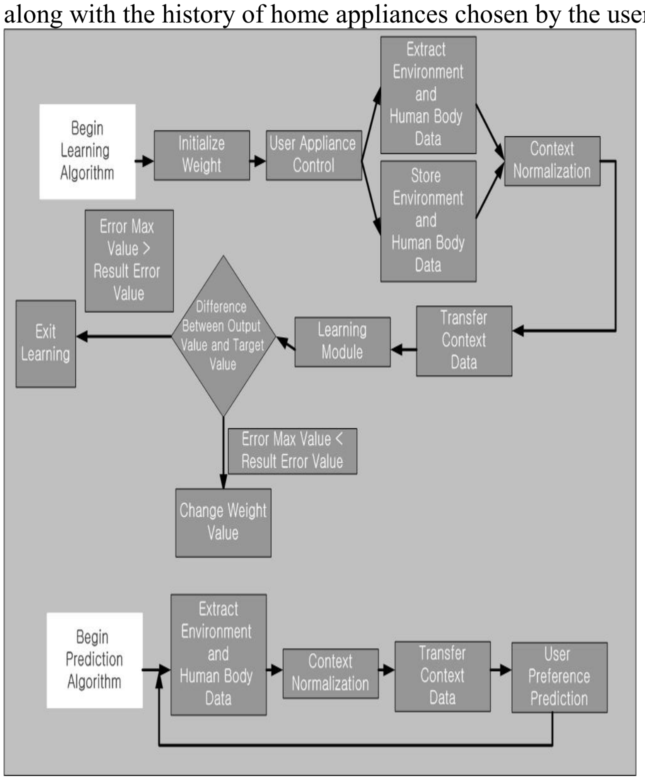


Fig. 7. User preference learning and prediction modules

Since back-propagation neural networks (NNs) have the capability to learn arbitrary non-linearity, we chose momentum back-propagation NNs for the user preference learning module [16]. Figure 7 shows the flow chart for the algorithm. For training and testing of the learning module, the difference between the trained result (context choice) and the user's actual choice is used to control the connection weights.

If the difference in value is smaller than the predetermined threshold, the learning process is completed.

4) User Profile Manager

The user profile manager maintains all history data about the user's home appliance choices. The user profile manager keeps the user profiles of the content choice and the corresponding context data. Figure 8 shows the XML schema structure of the user profile and an actual example of the XML data.

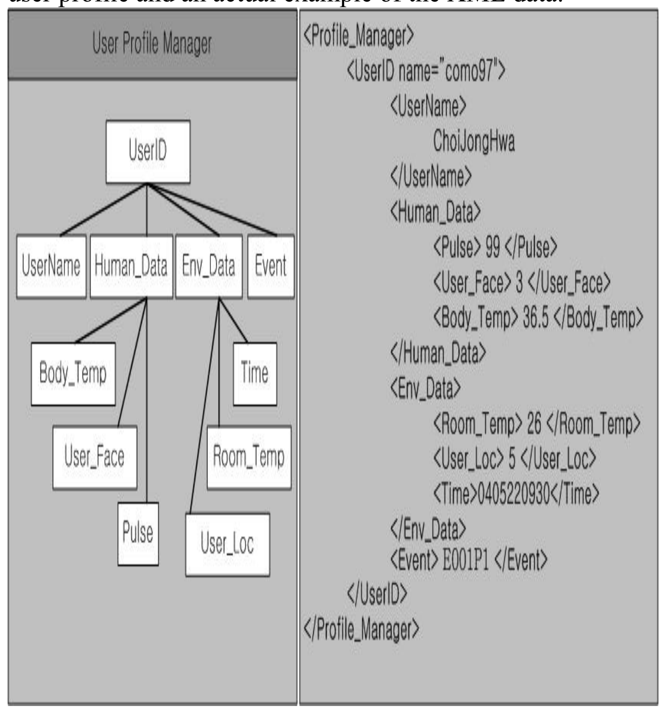


Fig. 8. Element Structure and XML schema of User Profile Manager

All context data are managed in the database. In case the machine-learning module losses a weight value, the user profile manager offers the machine-learning module all the state values that are stored in the database.

IV. Implementation and Experimental Results

The key modules of the context-aware middleware are the user preference learning and prediction modules. We experimented and evaluated different topologies of NNs. Variation of the topologies of the input layer, the hidden layer and the output layer was followed by measurement of error signal values by the hidden layer, error signal values by the output layer and the success rates (an error signal value means the summation of all error values at a specific layer) (in Figure 9). Variation of the number of trainings was followed by measurement of error signal values by the hidden layer, error signal values by the output layer and the success rates (in Figure 10). Variation of the number of neurons at the hidden layer was followed by measurement of error signal values by the hidden layer, error signal values by the output layer and the success rate.

As shown in Figure 9, the algorithm continues until the error signal value by the output layer is smaller than the predetermined threshold value. In each training experiment, one of the data groups was used the train the NNs, a second group was for cross-validation [17] during training and the remaining group was used to test the trained NNs. Figure 9 shows the definition of the values produced by the NNs.

The experiments show that a 6-3-3 topology (6 input units, 3 hidden units, 3 output units) has the best overall results, as indicated in Figure 9.

Topology	Learning Count	Success Rate(%)	Cross Validation error signal value by hidden layer	Cross Validation error signal value by output layer
6-3-1	69900	75	3.672139	25.172533
6-3-3	10922	100	2.775632	80.879432
6-3-5	54406	100	4.037638	160.923144

Fig. 9. Variation of the topologies

In Figure 10 shows that 30,000 trainings gave the best performance.

Learning Count	Success Rate(%)	Cross Validation error signal value by hidden layer	Cross Validation error signal value by output layer
1000	80	2.862910	76.219739
10000	100	2.863195	80.906453
20000	100	2.850291	75.866532
30000	100	2.877210	73.649298
40000	100	2.796571	78.721034
50000	100	2.790073	75.499323
60000	100	2.871029	76.502143

Fig. 10. Variation of number of trainings

The optimum number of neurons for the hidden layer is three, as shown in Figure 11.

Figure 12 presents output values in the learning and prediction module. Output values are separated into 5 state values.

The test results show that the pattern of an individual's preference can be effectively evaluated by adopting the proposed context model. Also, we have demonstrated the effectiveness of these algorithms in smart home.

Hidden Layer	Success Rate(%)	Cross Validation error signal value by hidden layer	Cross Validation error signal value by hidden layer
1	50	10.683244	10.683244
3	100	2.9223943	78.923310
5	100	3.1829323	128.920125

Fig. 11. Variation of number of neurons

Output Value → / Output Layer ↓	TV	Audio	Air Conditioner	Projector	Light
1	0.0	0.25	0.5	0.75	1.0
3	100	110	111	010	001
5	10000	01000	00100	00010	00001

Fig. 12. The definition of the output values

V. Conclusions

This paper described the context-aware middleware providing an automatic home service based on a user's preference at a smart home. The context-aware middleware utilizes 6 basic data values for learning and predicting the user's preference of the content. The six data sets construct the context model and are used by the context manager module. The user-pattern learning and predicting module based on neural network predicts the proper multimedia content for the user. The test results show that the pattern of an individual's preference can be effectively evaluated by adopting the proposed context model. Further research will be needed for adopting a different machine learning algorithm such as SVM(support vector machine)[18] and comparing the prediction ratio.

References

[1] Weiser. M, "The Computer for the 21st Century", Scientific American, September. 1991, pp. 94-104

[2] Sherif. M.H, "Intelligent homes: A new Challenge in telecommunications standardization", Communications Magazine, IEEE, Vol. 40, Issue. 1, Jan. 2002, pp. 8-8

[3] Kango, R., Moore, P.R., Pu, J, "Networked smart home appliances - enabling real ubiquitous culture", Proceedings of the 2002 IEEE 5th International Workshop on Networked Appliances, Liverpool, Oct. 2002, pp. 76 – 80

[4] Abowd, G.D., Dey, A.K., Brown, P.J. et al, "Towards a better understanding of context and context-awareness", Lecture Notes in Computer Science, Vol. 1707, Springer-Verlag, 1999
[5] Sun, J.-Z., Sauvola, J, "Towards a conceptual model for context-aware adaptive services", Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, Aug. 2003, pp. 90 – 94
[6] Anhalt, J., Smailagic, A., Siewiorek, D.P., Gemperle, F, "Towards Context-Aware Computing: Experiences and Lessons", Pervasive Computing, 2002
[7] Ranganathan, A., Campbell, R.H, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", Lecture Notes in Computer Science, Vol. 2672, Springer-Verlag, 2003
[8] Capra, L., Emmerich, W., Mascolo, C, "Reflective Middleware Solutions for Context-Aware Applications", Lecture Notes in Computer Science, Vol. 2192, Springer-Verlag, 2001
[9] Yau, S.S., Karim, F., Wang, Y., Wang, B., Gupta, K.S, "Reconfigurable Context-Sensitive middleware for Pervasive Computing", Pervasive Computing, 2002
[10] Open Service Gateway Initiative, Device Expert Group, http://www.osgi.org
[11] User Interface Markup Language, http://www.uiml.org
[12] Dobrev, P, Famolari. D, Kurzke. C, Miller. B.A, "Device and service discovery in home networks with OSGi", Communications Magazine, IEEE, Vol. 40, Issue. 8, Aug. 2002, pp. 86 - 92
[13] Crowley, J.L, "Context aware observation of human activities", Proceedings of the 2002 IEEE International Conference on Multimedia and Expo, Vol.1, Aug. 2002, pp. 909-912
[14] Charles, D, "The expression of the emotions in man and animals", Electronic Text Center, University of Virginia Library
[15] Chen, Z., An, Y., Jia, K., Sun, C, "Intelligent control of alternative current permanent manage servomotor using neural network", Proceedings of the Fifth International Conference on Electrical Machines and Systems, Vol. 2, August. 2001, pp. 18-20
[16] Weiss, S.M., Kulikowski, C.A, "Computer Systems that Learn", Morgan Kaufmann , 1991
[17] Burges, C.J.C, "A tutorial on support vector machines for pattern recognition", Data Mining Knowl. Disc., Vol.2, no 2, 1998, pp.1-47
[18] Burges, C.J.C, "A tutorial on support vector machines for pattern recognition", Data Mining Knowl. Disc, Vol. 2, no. 2, 1998, pp. 1-47

Jonghwa Choi received a B.S. in Computer Science from Sejong University, Korea, in 1986, an M. S. in Computer Science from Sejong University, Korea, Seoul, in 2005. He is currently an pursuing Ph.D degree at the Sejong University. His research interests are in Smart Home, Context-Aware Computing, Embedded System and Physics Engine for 3D Game.

Dongkyoo Shin received a B.S. in Computer Science from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently an Associate Professor in the Department of Computer Science & Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include XML Security, XML based middleware, digital right management for multimedia, mobile Internet and ubiquitous computing.

Dongil Shin received a B.S. in Computer Science from Yonsei University, Seoul, Korea, in 1988. He received an M.S. in Computer Science from Washington State University, Pullman, Washington, U.S.A., in 1993, and a Ph.D. from University of North Texas, Denton Texas, U.S.A., in 1997. He was a senior researcher at System Engineering Research Institute, Deajun, Korea, in 1997. Since 1998, he has been with the Department of Computer Science & Engineering at Sejong University in Korea where he is currently an Assistant Professor. His research interests include Mobile Internet, Computer Supported Cooperative Work, Object-Oriented Database, Distributed Database, Data Mining and Machine Learning.