2005－UBI－8（51）
2005／6／10

User Profile Management for Context-aware Applications in ubiHome Environment*

Youngjung Suh and Woontack Woo
GIST U-VR Lab.
Gwangju 500-712, S.Korea
+82-(0)62-970-2226
{ysuh, wwoo}@gist.ac.kr

ABSTRACT

Frameworks for context-aware services in a ubiquitous computing environment have been developed so far to allow the application services in the environment to monitor a sequence of patterns in user behavior as well as to manage the aggregated user profiles in one server. However, it must be a great burden for a whole environment to attend on the individual desires for the surroundings. Thus, we propose a user profile management framework in which the user's desires, needs, and preferences over services are managed through a wearable personal station, a terminal device which users have in wear. We exploit service-specific user preferences by describing them into context to provide personalized services. Also, we update the preferences dynamically changed by learning them. The details such as service properties or contents are able to be adaptively changed according to the learning results. The proposed framework can be widely exploited for ubiComp-enabling applications requiring the interaction between a user and an environment. Moreover, it may support personalized services within the range of protecting user's right to privacy.

Keywords: Context-aware, user profile, learning, personalized service

1. INTRODUCTION

Various kinds of context information are flowing in ubiquitous or wearable computing environments with a wide range of networking, computing, and distributed contents and services. Over the last few years, research activities on context and context-aware have been reported [1]. In these researches, there have been mentioned several kinds of contexts such as environment contexts, user contexts, computing resource contexts. Among those kinds of contexts, user-related context information is required to provide personalized services. Recent studies have made a great attempt to develop frameworks describing and utilizing various kinds of user-related context information to support the provision of personalized services. ClixSmart Navigator [2] implemented a Navigation Server between a content store and a WAP gate-way. As a similar system, Intelligent Software Agents Group of computing science department in University of Aberdeen developed a recommendation system called RECO [3] which aims to facilitate the use of context-aware services in a dynamic environment where users can access a variety of services from different locations. Meanwhile, there also have been researches on user profile data model to support context-aware service provision in the field of ubiquitous computing [4, 5, 6]. These are established through a Web interface. KDDI cooperation in Japan has been developing a framework for constructing common profile model and managing the personalized profile of user activities from mobile terminal devices [7]. In aggregating, updating and disseminating user profile, they implemented the framework as a client-server architecture in which Profile Aggregator called "Home Server" collects user profiles from a number of desktop PCs.

These approaches to researches on provision of context-aware services exploiting well-known user profiles have some limitations in that, they have the environment keep continuously monitoring a sequence of patterns in user behavior to provide users with context-aware services. They did not take user-centric context-aware service into consideration. In other words, in ubiquitous or wearable computing environments, users are likely to prefer spontaneously to disseminate his/her personal information within the desired situation rather than be monitored by the environment for enjoying context-aware services. Besides, those systems are restricted just to the level of recommending the expected user-desired service in sequence. That is, they do not support the personalized services in which the details such as service properties or

* This work is supported by Seondo project of Ministry of Information and Communication in Korea

contents are able to be adaptively changed according to a user.

To overcome limitations above, in this paper, we propose a framework in which wearable personal station (WPS) with a user and application services in an environment are the distributed independent entities, respectively, to interact with each other just when it requires exchanging a context between them. For this to result, we developed the mechanism to support providing context-aware services without a centralized home server. It can reduce a great burden for a whole environment to attend on the individual desires for the surroundings by managing the user's desires, needs, and preferences over services through a wearable personal station (WPS). Moreover, we exploit service-specific user preferences by describing them into context to provide personalized services. Then, we update the preferences dynamically changed by learning them. The details such as service properties or contents are able to be adaptively changed according to the learning results. Especially, in the proposed framework, it is considered for a user to take care of his/her personal information protecting his/her right to privacy. That is, users do need to distribute their profile only when they indeed want to enjoy services.

This paper is organized as follows. In section 2, we describe User Profile Management Framework. And the method for user profile management is explained in section 3. Section 4 deals with experiments. Finally, we conclude our work and give a brief remaining work in Section 5.

2. USER PROFILE MANAGEMENT FRAMEWORK

2.1 Context Model

To aware context information from user's activities in a daily life, a unified context is needed to create context for triggering services users want by analyzing and integrating information obtained from various kinds of sensors. Context used in the proposed framework is defined as a flexible but unified context describing a user's situation to trigger application services. In other words, it describes user's situation as who, when, where, what, how, and why (5W1H) and is shared between sensors and services [8]. We define different kinds of context, i.e. preliminary, integrated, conditional, and final context. The preliminary context generated from sensors is not enough to trigger an appropriate service. In other words, the preliminary context from a sensor may not be accurate or even incomplete since a sensor may not fill up all 5W1H, in general. Thus, we define integrated context, conditional context, and thus final context. The integrated context is obtained by integrating preliminary contexts from a set of sensors. We determine the final context which contains a set of parameters and a service function to be used to trigger a user-centered service. As a result, an application developer may easily develop context-aware application by specifying the condition of service to be triggered as a 5W1H of conditional context.

2.2 wear-UCAM

ubi-UCAM[8], Unified Context-aware Application Model, for providing context-aware services in a ubiquitous computing environment have been developed so far to allow the ubiServices in the environment to manage all user conditional contexts describing user's desires on a particular service. In addition, Interpreter in each ubiService provides graphical user interface to have users reflect their service-specific preferences by setting up user conditional context. To alleviate a great burden for all application services in a whole environment to care for the individual person's desires for the application services, we come to propose wear-UCAM [10], Unified Context-aware Application Model for Wearable Computing.

Especially, in wear-UCAM, the point that we would like to emphasize is to allow users to take care of his/her personal information, protecting his/her right to privacy. Thus, we make it possible that the user's desires, needs, and preferences over services are managed through a wearable personal station. wear-UCAM is a framework which offers personalized services to users according to service-specific user preferences by analyzing various kinds of contexts obtained from ubiSensors in an environment or wearSensors which users have in wear. The Figure 1 shows the overall architecture of wear-UCAM.

The main features of wear-UCAM are as follows. 1) updating user profile based on User Conditional Context, 2) obtaining high-level context analyzed through primitive context extracted from biological sensors which users have in wear, and 3) protecting user's right to privacy from an environment. Although wear-UCAM leverages the communication between wearSensors and wearServices by utilizing unified context as same in ubi-UCAM, it is able to be distinguished from ubi-UCAM in that privacy protection is supported by exploiting a user profile management technique. Ultimately user profile management framework can be cooperated with ubi-UCAM.

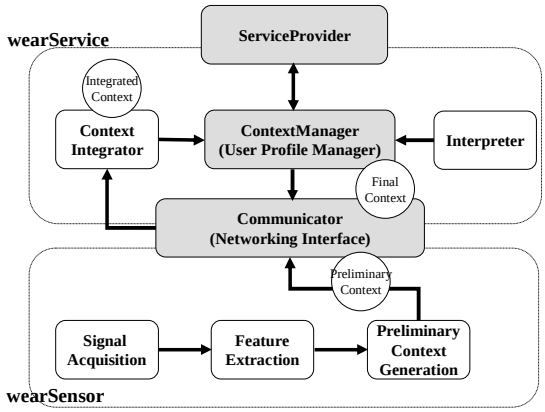

Figure 1. wear-UCAM

.

3. USER RPOFILE MANAGEMENT METHOD

The one of the important components for wear-UCAM is User Profile Manager. It is an extensible and featured version of Context Manager in UCAM, as shown in Figure 2.

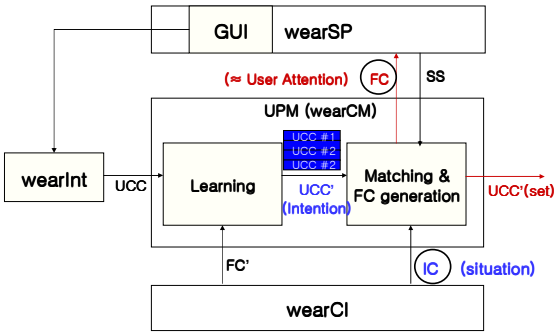

Figure 2. User Profile Manager in wearService

3.1 Context-based User Profile

User profile can be categorized into two according to its characteristics: one is static, the other is dynamic. Static user-related information is personal information such as name, age, address books, etc. Also, information which a user can initially set as his/her service-specific desires through GUI offered by Interpreter is considered as static. In case of the dynamic user profile, there can be the meaningful context information integrated from preliminary contexts describing user's biological conditions such as stress level, attention status, etc. These are obtained from biological sensors which users have in wear. Next, user's feedback information on services, service-specific user preferences, is likely to be changed dynamically. wear-UCAM shares unified context with ubi-UCAM. Thus, user-related information, user profile, is systematically and extensively described into each field of Context5W1H in the unified context [8].

3.2 User Profile Construction

As mentioned in section 3.1, user profile can be categorized into two according to its characteristics: one is static, the other is dynamic. Static user-related information is personal information such as name, age, address books, etc. Also, information which a user can initially set as his/her service-specific desires through GUI offered by Interpreter can be considered as static. These static user profiles are described in 5W1H of User Conditional Context, 'Who' being as a focal factor. Then, the dynamic user profiles are described in Context5W1H of Final Context since Final Context contains the results of service execution which a user desires. Thus, we can acquire the service-specific user preferences which are changed dynamically by collecting Final Context from ubiServices and wearServices. Integrated Context is also one of meaningful contexts describing user's biological conditions such as stress level, attention status. Finally, we can construct User Conditional Context which reflects user's indirect intention to enjoy a specific service, so that users can be provided the personalized services in ubiquitous/wearable computing environments.

3.3 User Profile Update

Although users new in a home environment can input their personal information and service-specific preferences through Graphical User Interface offered by Interpreter in wear-UCAM, it is needed to automatically update user's service-specific preferences when they are changed without any explicit user's command. And what if a user wants to enjoy a service which he/she has not initially set his/her preferences on it through GUI? In case of these situations, it is also needed to learn the user's behavior patterns to infer service-specific preferences, so that results of learning can be reflected into the dynamic update of user profile. The overall procedure of user profile updating is shown in Figure 3. First of all, we construct CUserProfile class from static user-related information inputted through GUI of Interpreter. Then, Final Contexts are aggregated at a periodic interval and stored in Context Data Base. At a

regular pace, we analyze and learn the dynamic user-related information from Final Contexts, referring to CUserProfile class. At last, we make a new User Conditional Context and send it to other ubiServices and wearServices.

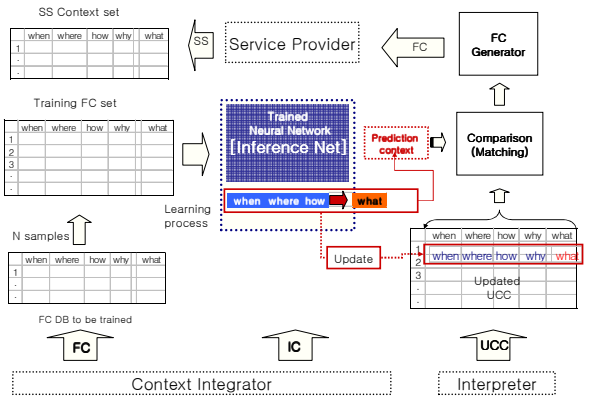

Figure 3. User Profile Update

In updating user profile, learning engine expects the proper service for given input. The input represents current situation of user's surroundings. We applied Neural Network to the learning engine. The context can be the training data patterns. As shown in Table 1, they can be generated from the collected Final Contexts, in the form of 5W1H. After a learning process by the training data patterns, an inference network can be constructed. Thus, we can infer the 'what' of 5W1H in Integrated Context. As I mentioned earlier, 'what' represents service identity, property, and contents which users want to enjoy in the specific situation. That is, learning engine outputs the proper result which represents service information. The history of service-specific user preferences, as the learning results, is stored in the context DB, so that it can be exploited for learning change of user's demands. Learning engine analyzes the history and determines new rules according to the learning result through a neural network.

Table 1. Final Context for Learning

Feature	Information
Who	A user who currently uses a service
When	Time when users enjoy serivices
Where	User's location
How	User's behaviors
What	Service name and properties

4. EXPERIMENTS

We have implemented 'ubiHome', a testbed for applying ubiComp enabling technologies to home environments. In ubiHome, various sensors are pervasive in order to generate preliminary contexts describing residents and their surroundings, and personalized services are offered by exploiting the resident's context. Each sensor is individually connected to a PC and does the work as a smart sensor with inherent processing, networking, and sensing abilities. We performed the experiments by using resources in ubiHome.

For a dynamic update of user profile, we mentioned that we applied a neural network to the learning engine of User Profile Manager. After getting the context (3W1H), we adopt the neural networks (NN) to induce an intention of a user from input contexts. It has long been postulated that neural networks might provide the established basis for approximating any (linear or nonlinear) function with a finite number of discontinuities. In particular, multi-layer perceptron (MLP) [11] might provide the most valid way to map any nonlinear relation, given sufficient neurons in the hidden layers, as shown in Figure 4. After proper training on a representative set of input and output vectors, MLP tends to lead a new input vector (that the MLP has never seen) to a similar output (to the correct output for the close input vector used in training)

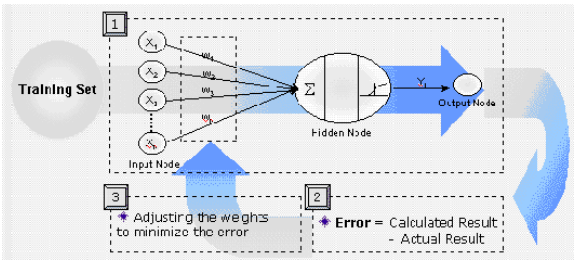

Figure 4. Multi-layer Perceptron

The adopted MLP consists of three layers including input, hidden and output layers, as shown in Figure 5. The adopted MLP is performed in two steps, i.e. learning and then mapping. First, MLP learns the nonlinear relationship between the context information (3W1H) and the user's intention ('what') through examples, rather than complicate behavior pattern analysis and mapping.

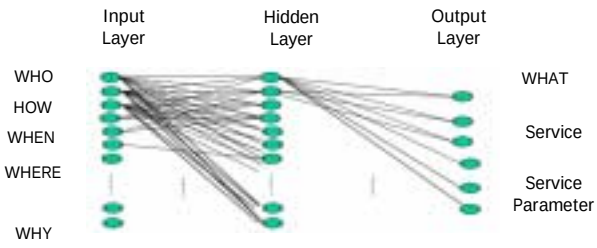

Figure 5. Organization of MLP

After a proper training, the MLP associates the input vectors with specific service/service parameters category, i.e. categorizes the user's contexts to several service categories. In this framework, a set of user's context (i.e. when, where, how, and why) are mapped directly to a set of symbolic representation of services i.e. {TV/Audio On/Off, etc.}, as input and output vectors of the MLP. Figure. 6 shows mapping sets of each layer in MLP.

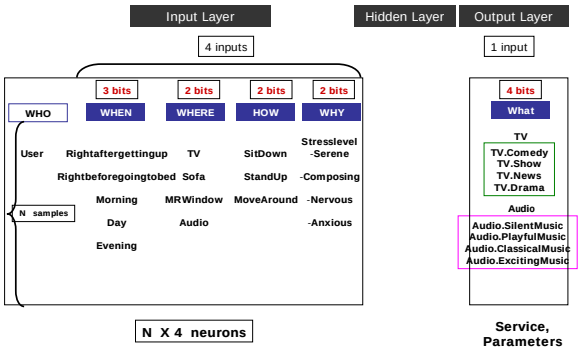

Figure 6. Data sets of each layer in MLP

Note that alternatively, the proposed User Profile Manager can be used to make the analysis procedure even simpler. For example, without extracting directly user's status and desire from physical sensors, context information {when, where, how, why} obtained from context-aware application model can be directly connected to the input layer of NN. Each context is denoted by numerical values, which stand for where the user is, when he/she is observed, which gesture he/she takes, and what his/her stress level is. Thus, these contexts can be trained to be mapped to the proper output categories. We described the experimental result in more detail as follows.

Inputs	4	4	4	4	4	4
Outputs	1	1	1	1	1	1
Vectors	160	160	160	160	160	160
Neurons	9	13	13	15	17	19
Hidden Layers						
the # of hidden layer	1	1	2	2	2	2
the # of nodes/h	4 (= INPUT)	8 (= INPUT X 2)	4 (= INPUT)	5	6	7
the total # of nodes	4	8	8	10	12	14
Training Parameters						
learning rate	0.3	0.3	0.3	0.3	0.3	0.3
Stopping condition						
the # of iterations	100000	100000	100000	100000	100000	100000
Tolerance %	95%	95%	95%	95%	95%	95%
Results						
total error rate	23.2131	19.466	21.1334	6.753	6.0425	3.7278
% Correct	11	7	8	23	21	42

Figure 7 – a . Learning Results with Network status (adjusting the topology of hidden layer)

We indicate inputs, outputs, vectors, and neurons as Network Status in Figure 7. In addition, hidden layer topology, training parameters, stopping condition and results are presented above. There is learning rate as training parameter. The number of iterations and error tolerance are included in stopping conditions. We can find that as the number of hidden layers is increased, the total error rate is decreased, other things being equal, as shown in Figure 7-a. Also, as the number of nodes within the hidden layer is increased, the total error rate is decreased. When the total number of nodes is fourteen in two hidden layers, correct percentage is at around 42%, which shows best performance with same experimental condition. In Figure 7 – b, it is shown that as we increase the number of iterations, others being equal, the total performance is improved, gradually. Comparing between 12 nodes and 14 nodes in two hidden layers with 3,000,000 interations, latter is better than former. T14 is found as the best one of results which we got from this experiment.

T7	T8	T9	T10	T11	T12	T13	T14
4	4	4	4	4	4	4	4
1	1	1	1	1	1	1	1
160	160	160	160	160	160	160	160
17	17	17	17	17	19	19	19
2	2	2	2	2	2	2	2
6	6	6	6	6	7	7	7
12	12	12	12	12	14	14	14
0.1	0.01	0.3	0.5	0.3	0.3	0.3	0.3
100000	100000	500000	500000	3000000	100000	1000000	3000000
95%	95%	95%	95%	95%	95%	95%	95%
6.0634	16.84	5.4123	12.0368	3.5412	3.7278	2.2381	1.3058
31	11	30	15	48	42	55	72

Figure 7 – b. Learning Results with Network status (adjusting learning rate and iteration number)

Figure 8 shows the results of comparing between actual outputs of prediction results and desired outputs of training sets. We can realize that the actual outputs seem to be approximated to desired outputs, as we expected.

Actual Output	Disired Output	Actual Output	Disired Output	Actual Output	Disired Output
13.2572	13	4.638171	5	3.323117	3
15.77974	16	4.877563	5	3.330195	3
1.025283	1	5.059554	5	2.806404	3
0.928683	1	4.964586	5	3.016124	3
0.9349977	1	7.048586	6	3.064165	3
1.133425	1	9.078062	9	2.987198	3
4.857995	5	12.18801	12	8.150376	8
5.024797	5	14.99448	15	10.6666	11
4.958493	5	4.835256	5	14.10716	14
5.039969	5	4.899389	5	16.31325	17
4.005897	4	4.98481	5	7.906893	8
4.062534	4	4.993693	5	10.65116	11
4.033796	4	1.254928	1	14.2172	14
3.943906	4	1.079246	1	16.34376	17
3.905409	4	0.896942	1	7.15222	7
3.988985	4	0.9161988	1	9.732225	10
4.013333	4	3.100619	3	13.56354	13
4.00633	4	3.117309	3	16.1084	16
6.651451	6	2.979155	3	7.089486	7
8.900903	9	3.053519	3	⋮	⋮
12.14961	12	2.863694	3	⋮	⋮
14.86247	15	3.148607	3		

Figure 8. Comparison btw actual outputs and desired outputs (Total Error Rate 1.3058, iteration # 3000000)

Figure 9 shows the Error Distribution Chart of an output variable. Through this chart, we can analyze which sample data of training sets affects to result in errors.

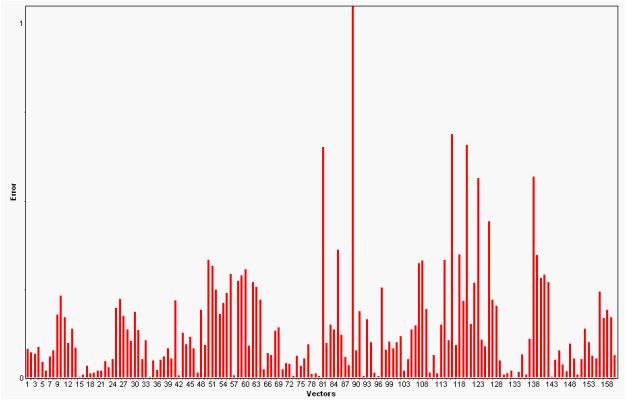

Figure 9. Error Distribution Chart
(Total Error Rate 1.3058, iteration # 3000000)

5. CONCLUSION AND FUTURE WORKS

In this paper, we designed and tested User Profile Manager using learning mechanism of Neural Network. Without extracting directly user's status and desire from physical sensors, context information {when, where, how, why} obtained from context-aware application model can be directly connected to the input layer of NN. Each context is denoted by numerical values, which stand for where the user is, when it is, which gesture he/she takes, and what his/her stress level is when he/she is observed. Thus, these contexts can be trained to be mapped to the proper output categories. The final result after learning using MLP is pretty good to recognize user's behavior pattern which can be used to infer the preferred service of the user. Remaining work is to consider the method which can reflect users' feedback in real-time. Also, even when any one of input contexts does not input to the system, we should make the system to provide the proper service to the user. We need to evaluate learning mechanism of Neural Network and consider a bayesian approach to human activity recognition [12]. On the other hand, it is a critical issue that users have to be able to enjoy personalized services by selectively disseminating their private information within the range of protecting their right to privacy. That is, users have only to distribute their profile only when they indeed want to enjoy services. Privacy control mechanism should be elaborated from now on.

6. REFERENCES

[1] Anind K. Dey and Gregory D. Abowd, .Towards a Better Understanding of Context and Context-Awareness., Proceedings of the CHI 2000 Workshop on .The What, Who, Where, When, and How of Context-Awareness., The Hague, Netherlands, April 1-6, 2000.

[2] Barry Smyth and Paul Cotter, 'Personalized adaptive navigation for mobile portals', *ECAI 2002*, (2002).

[3] Context-Aware Personalised Service Delivery. In R Lopez de Mantaras & L Saitta (ed), Proceedings of the Sixteenth European Conference on Artificial Intelligence - ECAI-2004 (Valencia, Spain), IOS Press, Amsterdam, pages 1077-1078, 2004.

[4] Ubisworld, http://www.u2m.org/

[5] D. Huynh, D. R. Karger and D. Quan, "Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF," In *Proc. of Semantic Web Workshop*, May 2002.

[6] SOUPA, http://pervasive.semanticweb.org/

[7] Daisuke Morikawa, Masaru Honjo, Akira Yamaguchi, Masayoshi Ohashi, KDDI Corporation, "A Proposal of User Profile Management Framework for Context-Aware Service," In SAINT-W'05, January 31 - February 04.

[8] S. Jang, W. Woo, "ubi-UCAM: A Unified Context-Aware Application Model," LNAI (Context03), pp. 178-189, 2003.

[9] Y.Oh, W.Woo, "A unified Application Service Model for ubiHome by Exploiting Intelligent Context-awareness," UCS'04, pp. 117-122, 2004.

[10] 홍동표, 우운택, "wear-UCAM: 착용형 컴퓨팅을 위한 컨텍스트기반 어플리케이션 모델," JCCI 2005, pp. 000-000, 2005.

[11] Richard P. Lippmann. An introduction to computing with neural nets. IEEE Acoustics, Speech and Signal Processing Magazine, 4(2):4--22, April 1987.

[12] A. Madabhushi and J. K. Aggarwal. A bayesian approach to human activity recognition. In *Second IEEE InternationalWorkshop on Visual Surveillance (CVPR Workshop)*, pages 25–30, Fort Collins,CO, June 1999.