JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 10, NO. 6, JUNE 2007(pp. 799-806)

Modeling of Context-aware Interaction in U-campus Environment

MoonWon Choo[†], YoungMee Choi[††], SeongAh Chin[†††]

ABSTRACT

The prototypical smart environment to support the context-aware interactions between user and ubiquitous campus environment based on multi-agent system paradigm is proposed in this paper. In this model, the dynamic Bayesian is investigated to solicit and organize agents to produce information and presentation assembly process in order to allocate the resources for an unseen task across multiple services in a dynamic environment. The user model is used to manage varying user constraints and user preferences to achieve system's goals.

Keywords: Ubiquitous System, Context-aware Environment, Multi-agent System, user Modeling

1. INTRODUCTION

The ubiquitous computing gives to the users the expectation that it is possible to access information and services anywhere. Besides that, the mobility provided by the ubiquity makes the user context, like the location, people and objects around, become more dynamic. The great variety of situations in which the user can be involved makes necessary a way to the applications adapt themselves according to the situations, providing a better support to the human-computer interaction. A way to improve the support for the human-computer interaction is to improve the communication during the interaction, making the computer able to process the contextual information of the user, the machine and the system communication, allowing the implementation of more useful computational systems. Context-aware applications use environmental context inputs to provide information to the user or to enable services for user[1,2]. The ways to use contextual information can be found in many researches. The DRAFT system investigates the traditional decision theory, adapting to a user's resource constraints, resource priorities, and the content priorities in a dynamic environment[3]. In this work, agent decisions are guided by a multi-attribute utility function as part of a user model. The issue is how to set up utility function dealing with many attribute efficiently and robustly. ITV system[3] suggests the online service modeling of context-aware interaction using rule-based approach. It may not be easy to model user preferences about services in advance precisely. In this paper, by using dynamic Bayesian, the prototypical model, SCUS (Sungkyul Campus Ubiquitous Service) to support the context-aware interactions between user and ubiquitous campus environment based on interface agent paradigm is proposed. In this model, the user may delegate tasks to a personal agent that may operate directly on the application or may act in the background while the user

※ Corresponding Author: MoonWon Choo, Address: (430-742)#147-2 Anyang-8 Dong, Anyang-City,
TEL: +82-31-467-8178, FAX: +82-31-449-0529,
E-mail: mchoo@sungkyul.edu

Receipt date : Dec. 13, 2006, Approval date : Mar. 19, 2007

[†] Division of Multimedia, Sungkyul University
[††] Division of Multimedia, Sungkyul University
(E-mail: choiym@sungkyul.edu)
[†††] Division of Multimedia, Sungkyul University
(E-mail: solideo@sungkyul.edu)

※ This work has been partially supported by Korea Research Foundation Grant reference KRF-2005-081-00012.

is doing something else. An agent is a representative of the user and acts on an user behalf more or less autonomously[4]. Moreover, it has to be able to communicate to the user in an appropriate way without being too much intrusive according to the context situation, user preferences, habits, needs and emotional states. SCUS (Sungkyul Campus Ubiquitous Service) is a prototypical smart environment implemented by adopting conventional multi-agent system and JADE platform[5]. The rest of paper is organized as follows. In section 2, the conceptual SCUS architecture is described. Then the context-aware interaction model is detailed in section 3 and 4. Section 5 shows the simulation and its results. We conclude with a discussion of the issues related to this model and future work.

2. SYSTEM ARCHITECTURE

The prototypical SCUS architecture based on the ITV model[6], is currently composed of three main conceptual components: the server, the network and the interface. The server concerns about how to store and to deliver the services and can to deliver different service objects to different users according to the users' characteristics and needs. The server must store the service objects available and the structure of the server programs that relates the objects inside the program in time, space and relationships. The content objects can be provided by various media types such as text, graphics, audio and video. The necessary semantics to increase the interactivity and facilitate how to get contextual information about the program to adapt the server services should be provided. The network needs to be flexible enough to adapt to the traffic conditions while keeping the presentation quality manageable[7]. The interface between the user and the system needs to be able to receive and process context-awareness information, and negotiate dynamically, with the server and with the network, the requirements and status of the application. The interface also may exhibit the presentation, composed of service objects which could be simple HTML/XML files or other multimedia objects . In this paper, we only focus on the middleware levels concerning context-aware interaction.

In ubiquitous campus applications, context-awareness support can be split into several categories[8]. First, individual service objects should be controlled dynamically. Those objects could be the official documents, multimedia lecture notes, message from library or simple pop-up messages for a particular seminar. Secondly, the service applications should adapt dynamically to the network and server conditions. Also the applications should dynamically adapt to user context. Depending on the parameters such as user identity, location and time, some services may or may not be offered. The methodology of personalized ubiquitous interaction is speculated by describing the entities composed of SCUS model shown in Figure 1.

3. MODELING CONTEXT-AWARENESS ENVIRONMENT

Context model allows the awareness of context in different situations as shown in Figure 1. Service agent can collaborate, via information interchange and services with user agent and service model, to maintain complex service levels.

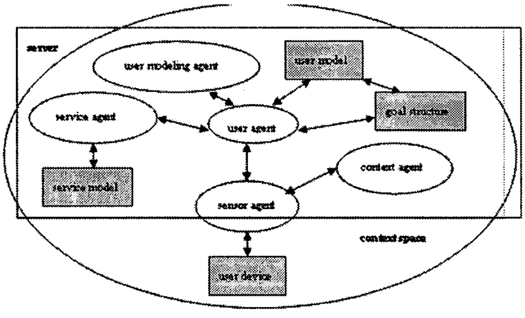

Fig. 1. SCUS model for context-aware interaction.

User own schedule of actions. Properties like reactivity, agent has its own ideas on to realize tasks and its adaptation, pro-activity, autonomy, mobility and social capacity are relative at the environment in that agents are inserted. The user-service agent interactions may represent the interactions between user, via several kinds of interfaces embedded on user device, and all applications of the environment. The context features defined by these interaction types can be represented in an XML structure compliant to the context ontology.

Context-aware aspects can be relevant when associated to environment, in which the service application is inserted, providing information for adaptability. Considering the execution of automatic services, it is possible to use the four entities: location, identity, activity and time. This is the key-point for defining concept of service model in this paper. Five arguments, who, where, how, when and what denote semantics that can be large complex. The semantic How, can be used for referring to what and how the resources being used by user. Therefore, implementing one evaluator mechanism for monitoring and showing a list of allocated resources, can it to get an environment status, indicating how the user is feeling itself in the environment. The proposal for joining the semantics HOW and WHAT provides support at creation of the activity entity. This entity is used for concept of service model, once that it is need allocated resources for generating activity at any level of the environment. Software agents can be used for supporting to automatic tasks and actions execution, observing the entities defined by the conventional agent paradigms. The development of service models for interactive applications, like is the case of Interactive Ubiquitous Campus, is a way for defining categories that indicate the status of an entity in one service, as well as one way for getting information about the behavior of this entity a long of time. Thus, using service models joined to contextual information, a service application can make actions and reactions like the follows:

- The presentation of both information and services to the user
- The automatic service execution
- The association of the context and information for later recovery

Service agents access the service profiles relevant to currently active user goals. Each service profile is composed and presented by four service elements: schedule, summary, page and view. Each element has different characteristics and different needs. The summary is a synopsis about service and its constraints. The user can choose the lecture notes for downloading/viewing, using summary information. The schedule is the service time table. Using the schedule, the user can reserve the midterm date to be updated and can ask the updated academic record early. The page is composed of timeline, media, information element and description. Information element contains the hyperlinks to other information element of other pages. From of the timeline, media and its description it is possible to build different interfaces, or presentation modes for one service. One service needs presentation mode specification and there are many types of available media. The view element serves for this assignment and its characteristics can be mapped into layout, communication, context and interaction. The layout will have typical interface variables like size, color, element position and fonts.

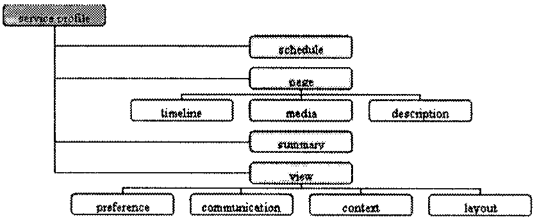

Fig. 2. Entities that compose the service profile .

The communication element describes all the communication requirements of the profile like bandwidth, plug-ins and QoS level. The context element informs all kinds of contextual information for one specified profile. This element is defined from context-awareness parameters like who, where, when, what, containing the actions that should be realized in determined situations. The interaction element concerns about which types of interaction are available to one profile. The user preferences can specify user's preferences related to content.

Each service can have many different service models. Models can differ in types of interaction, available media, aspects of layout, customization and privileges. Also this agent composes the each service entity for output using their XML metadata, for example. Adaptation to context features defined during the execution of user services can be applied at different levels: by selecting user's tasks from goal structure, by asking services in the environment to execute these tasks and by exchanging messages with the user. After selecting the task from goal structure, containing the description of the task and its constraints in terms of activation condition and priority, service agent execute the services under the help of user model. Context-awareness aspects are very important when associated with the environment in which the service application is inserted. These aspects refer it to operations associated to contextual sensing, contextual adaptation and contextual resources discovered that control the presentation of information and services to the user and the automatic execution of environment actions.

4. BAYESIAN USER MODEL

Both entities, User (agent) and the Environment, need to sense and elaborate context information. Given a task in the user's goal structure, once the user has been classified according to the strategy of the user model, its execution and results can be influenced by the context in which the interaction occurs. The context is characterized by the static and dynamic features defined by these two entities. Environmental features are described by service scope, type, physicality, etc.. In SCUAS, the following features are identified:

- Services: academic, interested group, circle activity, health, military, administration, etc.
- Type: public, private, don't care, etc.
- Physicality: PDA, PC, kiosk, etc.
- UserStatus: location, time, activity, emotional states, etc.
- Device: battery, connection, memory, etc.

User agent communicates with sensor agent, who communicates relevant changes to the context agent that knows the global context situation at the considered time. Sensor agent controls the user location and detects the user presence in the SCUAS and in particular its relative position to key places such as library, administration building, user common and classrooms. User agent also exchanges the messages with user model. Data in the user model are collected by direct user inputs through user devices, the derived information from the goal structure inserted by user and the information inferred by the environment. However, User modeling agent is a part of environment, starting modeling process when the user enters the environment. This model is utilized for adapting the service execution. Modeling strategy could be data-mining methodology or any reasoning approach. When user interaction is ceased, the inferred data are transferred to the user model for updating. Each user has his/her representative, implemented as User agent, may benefit of smart environment for the proactive execution of tasks that is scheduled in user's goal structure, which illustrates the tasks to be performed in different contexts and environments by User agent. When the user is situated in a particular environment, time, location, and emotional states, etc., UA is requested to perform personalized execution of services and to report the results appropriately. Therefore, user agent is to use several knowledge sources in interacting with smart environment according to user's tasks. These tasks may be explicitly requested by the user or inferred by the agent according to the level of autonomy. The User agent is modeled and implemented to have the reasoning behaviors aiming at the following goals.

- try to execute the user's tasks totally
- create new tasks if required in some context
- support the user model relevant for adapting task execution
- communicate results in personalized formats
- keep consistency between the user model and what has been inferred by an environment

In order to achieve these goals, several knowledge sources corresponding to sharing ontology used for inter-agent communication are to be exploited. In this paper, only the third goal is exploited. The user model's goal is to generate an assessment of users' interactions on the evaluation of past interactions in order to allow the user agent to provide tailored help that stimulates user learning. To generate its assessments, the user model keeps track of the user's behaviors during the interactions, since such behaviors are often a direct result of the user's preferences, or lack thereof. Modeling users' profile in interactive environment involves a high level of uncertainty and the Bayesian network gives the insight to solve this kind of problems[9]. Since there are many tutorial resources about the Bayesian approaches, the theoretic details are skipped here. Several random variables are introduced in the Bayesian network to represent user's behaviors and knowledge.

Context Nodes *E_x*: for each context object, the user model for that basic context object includes a node E_x that models a user's context to get an service expected. Each node E_x has two states: Sensitive and Ignored. In this work, $E_x \in$ [private, type, location, time, action, emotional states, view, battery, connection, memory]. The elements are the context variables, such as private/public, device types, user location, fixed/flexible scheduling, activity related, user's emotion, text/multimedia objects, and the rest are related to the network QoS respectively.

Emergency Nodes *EMS*: this node models a user's urgency to sense E_x to get an optimal service. This node EMS has two states: Emergent and Normal.

Status Nodes *Under, Grad, Alumni*: for each E_x, this node models a user's academic and graduation status to use optional services. The each node has two states: Applied and Not-Applied.

Click Nodes *Click*: this node models a user's action of selecting the services wanted. Each node has two states: Selected and Ignored. Selected denotes that the user has wanted the service manually, which could be the evidence node. The Service Node S_i denotes a service profile to be clicked and served. Figure 3 shows the basic dependencies among the nodes which encode the assumptions to be mentioned below.

The Conditional Probability Table(CPT) for each node is set up by taking the assumptions mentioned above into consideration. If there are multiple parent nodes for a particular node X, its probability table is defined in the following way: conditional assume that this node has n parent nodes. For each assignment of the parent node

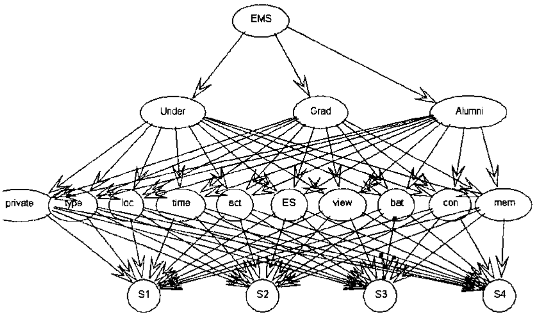

Fig. 3. The dependency between nodes.

values, if there are m parent nodes ($0 \leq m \leq n$) which have the state Ignored, for example, then the corresponding probability in the conditional probability table for this node to be Sensed is calculated using the following equation:

$$P(X = \text{Sensed}) = p - \{(p-(1-p)/n\} \times m$$

In this equation, p is designed by hand to denote a high probability as "Sensed". This equation generates the following CPTs:

1. If all the parent nodes are sensed (i.e., m=0), the probability of sensing X is p.
2. If all the parent nodes are ignored (i.e., m=n), the probability of sensing X is 1-p.
3. If 0<m<n, the probability of X being Sensed is between 1-p and p, and it decreases proportionally with the number of Ignored parent nodes.

The value of p is determined by considering the user's prior knowledge about the relevant knowledge domain. More direct evidence on user's selection are provided by user's interaction, such as clicking on a service shown on the menu on user's devices. The user model for a particular user's interaction must capture the unfolding of this interaction over time, and the corresponding evolution of the user's interactions. Traditionally, Dynamic Bayesian Networks (DBN) are extensions of BNs specifically designed to model worlds that change over time. DBN keeps track of variables whose values change overtime by representing multiple copies of these variables, one for each time slice, and by adding links that represent the temporal dependencies among those variables.

However, it often becomes impractical to maintain in a DBN all the relevant time slices. The rollup mechanism allows maintaining only two time slices to represent the temporal dependencies in a particular domain: the network at slice t-1 is removed after the network for slice t is established. An example of DBN is shown in Figure 4, where node C denotes evidence of a user's action at time t-1, while non-root nodes O_x and M could represent the context nodes in the network. Because user interaction can evolve with time, the nodes in time slice t must depend on the nodes in t-1. This can greatly increase the complexity of the corresponding probability tables. Consequently, the update of the networks can become quite time consuming. In this paper, an alternative approach to dynamically update the user model is used here. The following procedure shows how we update the model after an action C occurs:

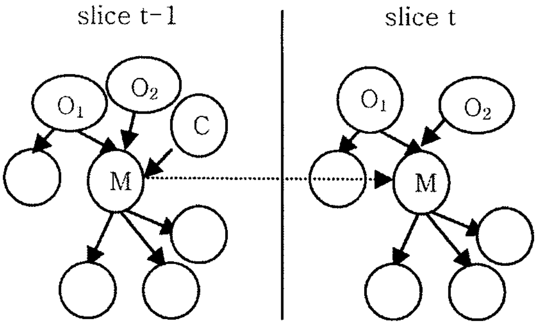

Fig. 4. Two slices of an example DBN.

After action C (i.e. Click)occurs, let us suppose with value true(=T), a new evidence node C is added to the network. After the network is updated, and before a new action node is taken in, the evidence node C is removed in slice t+1. The CPT for the node M is changed according to the value of C in slice t as follows:

$P(\text{M=T}|\text{C=T, } O_1\text{=F, } O_2\text{=F}) = P(O_1\text{=F, } O_2\text{=F}|\text{M=T}) + d,$

$P(\text{M=T}|\text{C=T, } O_1\text{=F, } O_2\text{=T}) = P(O_1\text{=F, } O_2\text{=T}|\text{M=T}) + d,$

$P(\text{M=T}|\text{C=T, } O_1\text{=T, } O_2\text{=F}) = P(O_1\text{=T, } O_2\text{=F}|\text{M=T}) + d,$

$P(\text{M=T}|\text{C=T, } O_1\text{=T, } O_2\text{=T}) = P(O_1\text{=T, } O_2\text{=T}|\text{M=T}) + d,$

where d is the weight the action C brings to the assessment of the corresponding context nodes, and it can be either positive or negative, depending on the type of interaction. That is, if the action Click reflects the negative meaning (such as

Ignored), *d* may has the negative value. If the M represents the context nodes E_x, the increment weight d_m is determined as follows.

$$d_m = \frac{1}{n}\sum_{i=1}^{N} p_i - p_m ,$$

where p_m is the probability of the node M and p_i is the probabilities of the occurrences of number *i* which are computed by applying the context M. The other types of nodes may take the heuristic increment weight values. The probability of context nodes M not directly affected by Click node don't remains unchanged, because we could be assumed to slowly loosen the unfired interactions. The S-shaped membership function is utilized to guess the weight to be applied to decay the unfired interaction using the following equation.

$$f(x) = \frac{1}{1+e^{-\alpha x}}$$

α is the coefficient for controlling the slope. The range of *x* can be scaled as 0 to *s*. The scaling weight is calculated as follows:

$$\beta = f(P(\cdot \mid M = F)/s),$$

where *s* is scale factor for *f*(*x*) and *d* is the given weight mentioned above. Hence, a node M may be decayed as following.

If P(M=F|OM=T) > θ, then P(· |M=F) = P(· |M=F) +β *d*,

θ is the given threshold value for making decision of decay and is used to determine the timing for Help event, which is not mentioned in this paper.

5. SIMULATION

Figure 5 shows the result of simulation which is set up for the test of plausibility of overall performance effectiveness. The main goal of simulation is to show that a user agent that could provide the appropriate services in the context of user's interaction pattern. Also we want to determine whether the user model suggested made reasonable assessments. The 200 simulation sessions are set up and the related parameters are determined such as d=0.01, θ=0.05, α=1.0, s=20. The resultant transitions of several CPTs after 200 simulation sessions reflect relatively correct assessment of the suggested user model. Three CPTs, P(S1=T|EMS=T), P(Under=T|EMS=T), and P(type=T|EMS=T) are calculated by the manipulated interaction data. In fact, this result has no practical significance, but it shows that emergent service could be more predicted to undergraduate students than other grades. And S1 service is more required than any other service as emergency, etc. based on simulation data.

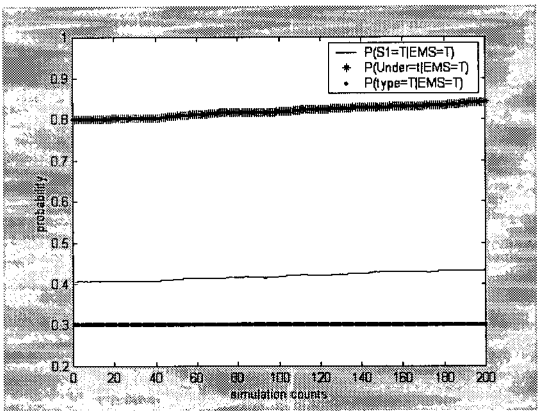

Fig. 5. +: P(S1=T|EMS=T), *:P(Under=T|EMS =T), and -: P(type=T|EMS=T).

6. CONCLUSION

There has been growing efforts to present a modeling of interactions of the system with the environment inserted in a context-aware and ubiquitous computing. Effective ubiquitous interaction requires a continuous modeling of both the user and the context. Therefore, ubiquitous computing system should be designed so as to work in different situations that depend on the factors stemming from presence of network connection, characteristics of interaction devices, user location, activity, emotional state and so on. The proposed modeling is the first step to help to determine and to formalize new types of interactions generated by a context-aware campus application. Towards to future campus, this modeling can be used to ease the development of environments, due to the fact complete interactions generated by user, network and application entities, are represented in the approach. These interactions are very important to context-aware, wearable, mobile and ubiquitous computing. However, more work in understanding how the user feedback influences the level of autonomy especially when his feedback is implicit. Also the detailed specifications for implementing working model is still under going.

REFERENCES

[1] Anthony Jameson," Modeling Both the Context and the User," *Personal Technologies*, 5, 2001.

[2] Dey A.K., "D-ME: Personal Interaction in Smart Environments," *LNCS*, Vol. 2702/2003, pp. 388-392, 2003.

[3] Terrence Harvey, Keith Decker, and Sandra Carberry, "DRAFT: Multi-Agent Decision Support Via User-Modeling," *AAMA'05*, pp. 25-29, July 2005.

[4] Jeffery M. Bradshaw, *Software Agents*, AAAI Press/The MIT Press, 1997.

[5] JADE: Web site: http://jade.tilab.com/.

[6] Jao Benedito and Rudinei G., "The Modeling of Structured Context-Aware Interactive Environments," *Trans. of the SDPS*, Vol. 5(4), pp. 77-93, 2001.

[7] David L.T. and David J.W., "Towards an Active Network Architecture. Multimedia Computing and Networking," *MMCN '96*, 1996.

[8] B. De Carolis, et. al, "A Personal Agent Supporting Ubiquitous Interaction," url:http://www.di.uniba.it/~ubium03/decarolis-1.pdf.

[9] Bellifemine, F. *et. al.*., "JADE A FIPA-compliant agent framework," *Proc. of PAAM'99*, pp. 97-108, 1991.

MoonWon Choo

received the Ph.D. degrees in computer science from Steven Institute of Technology, Hoboken, NJ, in 1996. He is currently Associate Professor in the Division of Multimedia at Sungkyul University since 1997. His main research interests include adaptive vision system and multimedia application.

YoungMee Choi

received Master in Mathematics at Ihwa University and Ph.D in Computer Science from Aju University, Suwon, Korea. Since 1994, she has been with the Division of Multimedia, Sungkyul University, Anyang, Korea. Her main research interests include AI, Software Agents and Multimedia applications.

Seongah Chin

received the Ph.D. degrees in computer science from Steven Institute of Technology, Hoboken, NJ, in 1999. He is currently Associate Professor in the Division of Multimedia in the College of Engineering at Sungkyul University, Republic of Korea. In 2001, He joined the faculty of Sungkyul University, Anyang, Republic of Korea. Prior to joining Sungkyul University faculty, he worked as a Research Professor in the Graduate School of Media Communications at Sogang University. His primary research interests focus on the development of creating virtual human in mobile devices as well as PCs particularly in enhancing the efficiency of collaborative environment.