A STOCHASTIC APPROACH FOR CREATING DYNAMIC CONTEXT-AWARE SERVICES IN SMART HOME ENVIRONMENT

Hua Si*, Yoshihiro Kawahara*,
Hiroyuki Morikawa†, Tomonori Aoyama*

Abstract
Context-aware service platforms aim at providing users with dynamic services that adapt to changeful environment. Though various toolkits have been developed, it is still not easy for end-users to program their own personalized services. In this paper, we present a context-aware service platform called Synapse that can adequately relate users' contexts and services, and automatically generate context-aware services based on users' habits. By exploiting the recorded histories of contexts and services, Synapse can learn different users' habits. Then Synapse can predict the most relevant services that users will use in current situation based on their habits, and provide services in different modes. As learning and predicting mechanisms are on the basis of a stochastic approach – Bayesian Networks [5], Synapse can absorb various uncertainties arising from sensor data and provide truly personalized services.

1. Introduction

Context-awareness is now regarded as a key ingredient for pervasive computing, and several toolkits such as Context Toolkit [8] and Location Stack [2] have been proposed to incorporate users' contexts into network applications. A category of attractive context-aware applications is desired in indoor environments such as home and office [5]. In such an environment, dynamic services enabled by the seamless integration of sensors, home appliances and networks will not only give great helps to the elders and patients, but also facilitate the daily life of ordinary people. Some prior projects focused on recognizing the contexts, such as users' activities and environmental conditions, from a large number of simple sensors [10]. Some other projects propose ECA Rule that allows application developers to specify and associate events with actions in a rule-specification language [3]. However, these works are not really easy for end-users to program their own personalized services. Therefore, a dynamic mechanism that can automatically generate context-aware services based on the users' habits is necessary. We are developing a context-aware service platform called Synapse, which can learn different users' habits by exploiting the recorded histories of contexts and services, and then predict and provide the most relevant services that users will use in current situation based on their habits.

The following "daily scenarios" are learned and implemented in a smart home test bed of Synapse:

- The "Light" scenario: if it is too dark in the room, Synapse will automatically turn on the light.
- The "TV" scenario: Synapse will recommend TV programs appropriate for people in living room. (When only kids are watching TV, cartoon program will be recommended. When parents

* Department of Information and Communication Engineering, The University of Tokyo
† Department of Frontier Informatics, The University of Tokyo

and kids are watching TV together, movies or some other programs will be recommended on the basis of user's favour.)

- The "Music" scenario: Synapse will automatically turn down the volume of music player when someone is using the phone, and turn up the volume after using the phone.

Our basic assumption of this system is that a user may repeatedly choose preferred services in similar situations, which are his habits. Obviously, the best grounds for learning users' habits exist in the recorded histories of services used by users and the contexts (we call "sensor events" in Synapse), which indicate the situations in which each service is used. Considering the complexity of human habits and the flexibility of the system, we choose one of Bayesian Networks [6], HMM (Hidden Markov Model) [7] to build the stochastic model of Synapse. This model consists of continuous cycles, and each cycle is composed of two phases: the Learning Phase and the Executing Phase. Firstly in a Learning Phase, Synapse learns the relationships between sensor events and services by exploiting recorded histories of them. Then in Executing Phase, based on the learned relationships and the current sensor events, Synapse predicts the most possible services to be used and provides them to users. Since users would like to enjoy the autonomous services in a moderate degree without losing control of them [1], Synapse provides services in two modes: Active Mode will start a service automatically, where Passive Mode recommends the top 3 relevant services in a list and let users select as they wish. The results of the Learning Phase are used as the prior knowledge for the next cycle. To easily achieve the personalization, the user ID is treated as a sensor event.

We will discuss the architecture and implementation details of Synapse test bed in section 2. The conclusions will be given in section 3.

2. Architecture and Implementation of Synapse

The test bed of Synapse consists of four parts: 1) the Synapse Core, 2) the sensors capturing real world information, 3) the home appliances providing services, and 4) the web user interface, as shown in Figure 1.

2.1. Synapse Core

We use HMM to model the relationships between sensor events and services. As shown in Figure 2, there are two basic components in HMM: the hidden states X_t and the observations of states Y_t. In Synapse, each hidden state X_t corresponds to a service S_t[*], to indicate the situation in which this service is used, and the observation is a vector of sensor events E[†] extracted from sensor data. In a Learning Phase (t=1, 2...T), we use forwards-backwards algorithm and EM (expectation maximization) algorithm to learn the parameters of the model ($\pi(i)$, $A(i,j)$, $P(Y_t|X_t)$)[‡]. After that, in an Executing Phase (t>T), we use the learned parameters and new sensor events to compute the occurrence probability of each service through a filtering algorithm and sort them. If a probability is higher than a threshold, the corresponding service will automatically start in Active Mode. The top 3 services will be recommended in a list to the user interface in Passive Mode. All these algorithms are implemented on Matlab.

---

[*] S (SN, ST): SN - service ID, ST - the time this service is recorded.

[†] E (EN, EV, ET): EN - event ID, EV - value of event, ET - the time this event is recorded.

[‡] $(i)=P(X_1=i)$ is the prior probability of HMM. $A(i,j)=P(X_t=j|X_{t-1}=i)$ is the transition matrix of HMM. $P(Y_t|X_t)$ is the observation model of HMM.

2.2. Sensor Event Collection Part

Sensor event collection part captures the real world information via various sensors, converts raw data into useful contexts (we call "sensor events" in Synapse), and records these sensor events in database. Sensor Aggregator fuses the raw sensor data and reduces the noise. For instance, the average temperature in a room is fused from different temperature sensors. Context Inference extracts the complex events such as "the user is sleeping" from simple events. In this test bed, 4 kinds of sensors are used to produce 11 sensor events: RFID is used to identify Users, U^3 [4] nodes are used to capture the temperature, brightness and human motion, a contact detector detects whether the phone is in use, and an e-calendar is used to detect a day of the week.

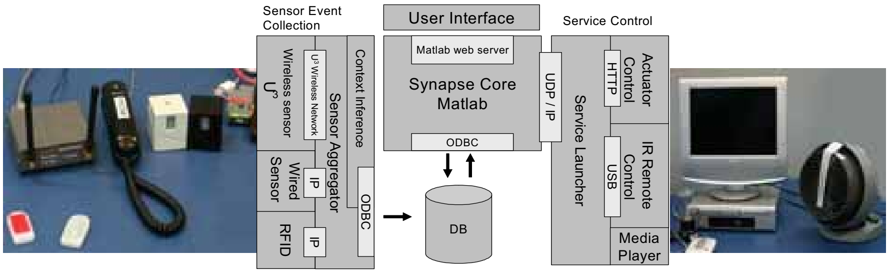

Fig. 1. Architecture of Synapse. Sensors are on the left, and services are on the right.

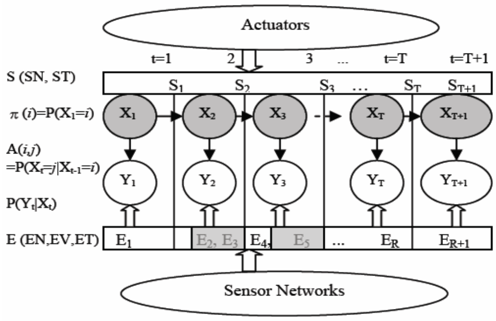

Fig. 2. Model of Synapse. The grey rectangles indicate a certain interval before ST.

2.3. Service Control Part

Service control part controls various home appliances to supply services. Service Launcher operates as a proxy between Synapse Core and the home appliances. It receives the service ID from Synapse Core through UDP/IP networks, and controls the home appliance corresponding to this service ID. It also sends the ID of a selected service to Synapse Core for history-recording. As a result, the scalability is ensured, since Synapse Core can manage the services only with IDs, and ignore the various operations of home appliances. In this test bed, 4 home appliances are used to provide 23 services: a light and a fan provide on/off services, a TV provides on/off, 12 channels and 2 videos, and a music player provides on/off and music mute/loud services.

2.4. User Interface

Synapse provides a user interface in HTML form on Matlab Web Server. Users can browse this webpage through PC, PDA, or cellular phone, and start a service only by selecting this service ID. The recommended service list on this website can automatically update after a fixed interval, or be manually updated.

3. Conclusions

In this paper, we presented a general context-aware service platform—Synapse. By exploiting the recorded histories of contexts and services, Synapse can learn the users' habits. After that, Synapse can predict the most relevant services that users will use in current situation based on their habits, and provide services in Active Mode and Passive Mode. We also described the architecture and implementation details of a smart home test bed of Synapse. The initial performance evaluation of Synapse can be found in [9]. In the future, we will extend the sensor and service parts of our test bed and build a whole Synapse system in a real house, and put the system under real inhabitants' tests.

4. References

[1] BARKHUUS, L., Is Context-Aware Computing Taking Control Away from the User? Three Levels of Interactivity Examined, in: Proceedings of Ubicomp 2003, LNCS 2864, (2003).

[2] GRAUMANN, D., Real-world implementation of the location stack: The universal location framework, in: Proceedings of WMCSA 2003, (2003), 122–128.

[3] IPINA, D. L., An ECA Rule-Matching Service for Simpler Development of Reactive Applications, in: IEEE Distributed Systems, Vol. 2, (2001).

[4] KAWAHARA, Y., Design and Implementation of a Sensor Network Node for Ubiquitous Computing Environment, in: Proceedings of IEEE Semiannual Vehicular Technology Conference, 2003.

[5] MEYER, S., A survey of research on context-aware homes, in: Proceedings of the Australasian information security workshop conference on ACSW frontiers, Vol. 21, (2003).

[6] MURPHY, K., Dynamic Bayesian Networks: Representation, Inference and Learning, PhD Dissertation, UC Berkeley (2002).

[7] RABINER, L. R., A tutorial on Hidden Markov Models and selected applications in speech recognition, in: Proceedings of the IEEE, (1989), 77(2):257–286.

[8] SALBER, D., The context toolkit: Aiding the development of context-enabled applications, Technical report, Georgia Institute of Technology (2000).

[9] SI, H., Performance Evaluation of a Context-aware Service Platform Synapse, in: Proceedings of IEICE General Conference, Osaka, Japan, (2005).

[10] TAPIA, E. M., Activity Recognition in the Home Using Simple and Ubiquitous Sensors, in: Proceedings of Pervasive 2004, LNCS 3001, (2004), 158-175.