A Context-Aware System for Smart Home Applications

Wen-Yang Wang[1], Chih-Chieh Chuang[1], Yu-Sheng Lai[1], and Ying-Hong Wang[2]

[1] Computer and Communications Research Laboratories,
Industrial Technology and Research Institute, Tainan City, Taiwan
{wwj, twjack, laiys}@itri.org.tw
[2] Department of Computer Science and Information Engineering, Tamkang University,
Taipei County, Taiwan
inhon@mail.tku.edu.tw

Abstract. Context-awareness is an important part for ubiquitous computing. Many applications of ubiquitous computing have to access some related contexts in order to provide the right services at the right time and the right places. However, there are some challenges for applications in ubiquitous computing, especially for those in smart home. These challenges make the design of smart home applications much more difficult than other applications. Therefore, we propose a context-aware system, CASSHA (Context-Aware System for Smart Home Applications), which is designed for smart home applications. CASSHA consists of the components for processing, representation, provision, and coordination, and is able to provide required contexts for smart home applications without conflicts. The system overcomes most of the technical challenges for smart home applications, and satisfies the requirements for these applications.

1 Introduction

In ubiquitous computing, most of the applications may need to collect several different contexts. The example of contexts may include user identification, positioning and tracking, user's activities, as well as time, temperature, et al. These contexts are required for the applications in the field of ubiquitous computing, especially for those in smart home. To make a "smart" decision, the applications may need much information of the user and the surroundings. Take the tour guide application as the example, the application needs to know the user's current location and his destination, and then points out a path for the user. The path may be merely the shortest path, or may be a less crowded one according to other users' activities and events in the environments. In other words, the later one will be the path that will have less people and traffics when the user passes through. In order to compute the less crowded path, the application needs to be aware of more contexts other than the source and destination of the current user. Thus, context-awareness is needed for applications of ubiquitous computing.

In order to collect more contexts, different types of sensors will be needed. For examples, thermometers measure the temperatures, while locators show the locations of users. However, the data generated by these sensors are simply raw data and cannot

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 298 – 305, 2005.

be easily used by applications. Besides, some contexts cannot be retrieved directly from single type of sensors, and may only be retrieved after several analyses. An example is user activity. If a user is sitting on the sofa in the living room, and the television is turned on, then it is possible that this user is watching TV. In this case, both user's location and the status of the TV are required to decide the user activity of watching TV. If more information is retrieved, including the user's preference, content of TV program, and user's viewpoints, then the analyzed result will be more precise. As we can see, to determine user activities precisely is too complex for a single application. Therefore, a context-aware system for providing and analyzing context is needed for ubiquitous applications.

For smart home applications, there are much more difficulties on designing the system. Infrastructure at home may not be sufficient for supporting new applications, devices may not be bought at the same time, no system administrator is available, and user activities are more difficult to predict at home than in other places. Besides, there are other social implications and marketing issues that also increase the difficulties for application in smart home. Therefore, designing context-aware systems for smart home applications is more difficult than systems for other applications.

In this paper, we summarize the challenges of smart home applications from previous researches, and design CASSHA as a context-aware system that can overcome, or at least mitigate, these challenges. Our design uses a layered approach, including Context Processing, Context Representation, Context Provision, and Application Interface. In each layer, functionalities are also defined in order to overcome or mitigate the challenges of smart home applications.

The rest of paper is organized as follow. Section 2 first addresses the challenges of smart home applications. And in section 3, previous works of context-aware systems are discussed. Afterwards, CASSHA is presented in section 4. Finally, section 5 concludes our work and discusses issues for future researches.

2 Challenges

As people can expect, smart home becomes the trend of future homes. Smart home is able to integrate technologies of digital living and provides a comfortable, secure, and convenient living style. However, smart home also introduces challenges for designing applications in this environment. Smart home applications differ from others in many aspects. Some researches [5] have been presented to address these issues and the impacts of smart home applications. These challenges includes:

1. **Seamless Platform.** There are various types of devices at home: multimedia servers, intelligent monitors, home appliances, as well as sensors and controllers. It is unpredictable that when and where these devices will be installed. In order to connect and maintain interoperability of these "accidental" devices, not only the standards of network transmission should be considered, but also the construction of a seamless platform is needed. With the seamless platform, various smart home applications can thus cooperate with each other.
2. **Mechanism for Open Service Management.** There will be many different smart home applications in the future, and the number of these applications will grow up as the time goes by. One the one hand, new applications should be able

to add into the smart home when they are presented. On the other hands, some old applications may be updated with more powerful functionalities. Therefore, there must be a mechanism for managing smart home applications, in order to dynamically add new applications and keep upgrading the old ones.

3. **Convenience and Reliable with No System Administrator.** Home users do not have enough knowledge and experiences of ubiquitous computing technologies, nor do they have a dedicated administrator to manage and maintain the applications at home. However, smart home applications should still be able to execute under these circumstances. Therefore, these applications must be convenience and reliable, even when no system administrator is presented.
4. **Inference under Various Requirements and Situations.** Home users may use smart home applications under different requirements and situations. Moreover, some away-from-home activities may also impact these applications. Therefore, applications must fulfill various requirements and situations, and provide expected results by inferences.
5. **User-oriented Designs.** There will be plenty of applications designed for different scenarios and use cases in smart home. However, these applications should be designed based on users' requirements, and should provide a convenient interface if user interactions are needed.

As we can see, the challenges of smart home applications are more difficult than other applications. In order to reduce the difficulties for designing smart home applications, a context-aware system that manipulates these challenges is needed; hence, the design of this system is much more difficult than other systems.

3 Related Work

Previous researches have introduced different approaches of the context-aware systems for different applications. Though the context-aware system for smart home applications is not a new research topic, former researches cannot fulfill all the requirements of these applications.

Lee and Chung proposed system architecture for context-aware home applications [9]. They classified home applications into several categories, and provided four scenarios for media, healthcare, control, and management applications respectively. After that, they analyzed the required technologies for each scenario, and proposed a structure of home application server that can integrate these technologies. However, this home application server saves only raw data in its database, and does not have a well-defined representation for contexts. These will provide less reusability of contexts and lead to the difficulties of making inferences.

Another context-aware system for home applications can be found in [10]. Universal Home Network Middleware (UHNM) uses Adaptors to achieve network interoperability, uses several managers to take care of applications, and uses Messaging Layer for communication between managers. It is designed to provide zero-configuration, high-level abstraction, context-awareness, and adaptation. However, UHNM does not have well manipulation of contexts; it only utilizes the data from sensors and cannot perform inferences from sensed data at all. Besides, UHNM also lacks of a database for recording the events and the sensed data.

Authors in [2] also proposed a context-aware middleware for controlling home appliances. The middleware is based on OSGi (Open Service Gateway Initiative) and uses UIML (User Interface Markup Language) to define user interfaces. This middleware focuses on the learning and prediction of user preferences, and uses neural networks to do the learning and prediction. However, only user preferences can be provided by the middleware, which may not be sufficient for all the smart home applications. Besides, due to the nature of neural networks, the training data should be large enough in order to make good predictions—this may cause inconvenience for users at home, and needs an experienced user to master the training process.

Still some related work takes different approaches and focus on other aspects. In [7], a Service-Oriented Context-Aware Middleware (SOCAM) has been proposed. The key feature of SOCAM is its ability to reason about various contexts. And in [8], an agent-based approach is used. Mobile agents travel between devices to collect data and perform activities of the applications. Both these two approaches do not have the mechanism to coordinate applications, and may lead to unpredicted results and reduce the reliability of the systems. Other systems such as [3], [4], and [6], are not designed for smart home applications. Most of these systems require infrastructure and administrators, which are not appropriate for smart home applications.

4 Context-Aware System for Smart Home Applications (CASSHA)

According to the challenges mentioned in section 2, we propose CASSHA for smart home applications. CASSHA is designed to overcome or mitigate these challenges of smart home applications. It is a seamless platform for the interoperability of various devices at home, provides an interface for applications to be managed, no administrators are needed, inferences can be made under various requirements and situations with its inference engine, and it is designed for usage of smart home applications. CASSHA includes four components in order to achieve these goals. These components are shown in Fig. 1, namely Context Processing, Context Representation, Context Provision, and Application Interface. We will discuss these components respectively in the followings.

4.1 Context Processing

Context Processing does the job to collect the raw data from sensors and to interpret contexts thereafter. It collects the raw data from all sensors at home. In order to achieve that, it bridges the physical networks, identifies the type of sensors, and reads the measurements from them. As mentioned before, these sensors may be bought at different time, and may be installed or replaced at some other time. Once a sensor is installed, Context Processing will be aware of these sensors through additional bridges and adapters. Then it identifies the type and the location of these sensors through some service discovery mechanisms (such as UPnP [12]) and localization methods. When a sensor is removed or unable to response, Context Processing then marks the sensor as unavailable. By these means, Context Processing is able to adapt the present and absent of the sensors.

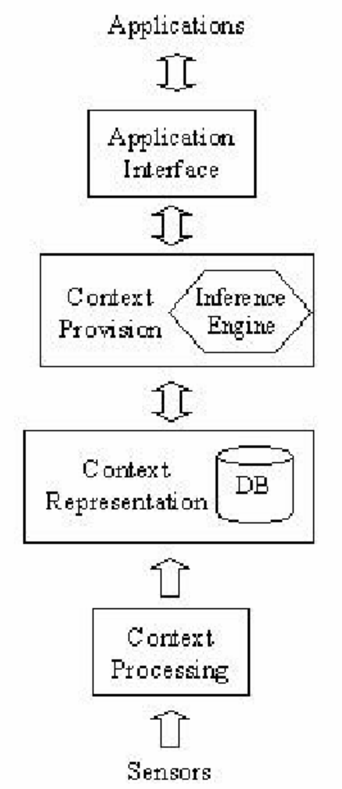

Fig. 1. The proposed context-aware system for smart home applications

After collecting the raw data from sensors, Context Processing then interprets contexts from these raw data. It will interpret the raw data according to the sensors' types, characteristics, and their locations. For example, the data read from the thermometer in the living room will be considered as the room temperature of the living room. In addition to the interpretations of simple contexts, Context Processing will also combine the related contexts. Contexts sensed in the same place and at the same time will be combined together, as they may be highly related to each other.

4.2 Context Representation

Context Representation normalizes the contexts collected from Context Processing and saves the normalized contexts into the database in order to query and analyze. It plays the role for improving the reusability of collected context. When receiving the context from Context Processing, Context Representation manipulates the contexts and translates them into a normalized representation. The representation may be ontology-based or in other forms [1, 11].

After normalizing, these contexts will be saved into a database. With this database, contexts can be reused for any required applications. Once contexts are saved, the applications can query contexts through Context Provision at any time, and Context Provision can also analyze these contexts in order to produce other high-level contexts. Take user tracing for example. In order to obtain the path that a user moves, previous contexts of the user's location will be needed and these contexts can be access from the database to enhance reusability. Another example is that for the learning mechanisms in Context Provision, former status of contexts may be needed, and these contexts can also be accessed directly from this database.

4.3 Context Provision

Context Provision provides the contexts for smart home applications. When an application requires some contexts, Context Provision acts as an interface to query contexts from the database in Context Representation. If the needed contexts are simple contexts and can be found in the database, Context Provision will soon return these contexts to applications. However, if high-level contexts such as user activities are needed, Context Provision will have to analyze the contexts in the database and provide the results to applications.

In order to perform the analysis, Context Provision has an inference engine to do the job. The inference engine takes several contexts, as well as user profiles and environment layouts into consideration. Therefore, the results of high-level contexts can be more meaningful than the contexts simply produced by Context Processing. In addition, with well-designed user profiles and a good learning mechanism, the user activities at home can become more noticeable, and the difficulties of analysis can also be reduced as well. For example, high-level contexts such as user activities can be obtained through the inference engine. Although some of early analyses of user activities may lead to unexpected results, with the learning mechanism and user profiles, the accuracy will be enhanced after several trials.

4.4 Application Interface

The main function of Application Interface is to provide an interface for smart home applications to access the system. This interface hides the lower layer details from applications. Applications only have to know what contexts they required, and do not have to know how to get these contexts from which sensors.

In addition to provide the interface, Application Interface has another important function as well—to resolve the conflict between applications. In normal situation, Application Interface is simply an interface for applications to access the system. However, when conflicts occur between two or more applications, Application Interface has to coordinate the conflicted applications. These conflicts may result from two or more applications designed for different situations. For example, an energy reservation application will turn off the air conditioner if no one is at home, but another application may turn on the air conditioner when users are arriving home. In this case, Application Interface must coordinate these two applications according to when will users arrive home. When the number of home applications grows up, the occurrences of such conflicts will become more frequently. Therefore, Application Interface plays an important role to resolve these conflicts.

4.5 Discussion

With CASSSHA, we can overcome or mitigate the technical challenges of smart home applications. CASSHA is indeed a seamless platform for smart home applications. The interoperability is achieved by Context Processing. Once devices and sensors are added, Context Processing will take care of the installation and configuration of these devices and sensors, and make use of them. Context Processing collects different type of sensors and shields the heterogeneity from upper layers. Besides, it can merge different physical networks at home with some additional adapters or bridges for each network.

As for the challenges of application management, Application Interface does a good job. The interface is provided not only to the applications at home, but can also be provided to service providers under user's permission. Therefore, service providers are able to use the interface to see what applications can be added in the smart home. Besides, old applications can be updated through the same mechanism.

When it comes to the challenge of convenience and reliable with no system administrator, all of these four components contribute to overcome this challenge. Context Processing provides the interoperability of devices and sensors, which may ease the installation progress for home users. Context Provision learns and analyzes user preferences, while Application Interface resolves the conflicts of applications. With these two components, administration is only needed when experienced users want to tuned the system for more detailed settings. And reliability is achieved with the assists of Application Interface and the database in Context Representation. Due to Application Interface resolves the conflicts of applications, thus the possibility of system failure will be reduced. However, if a failure still occurs, the database in Context Representation provides capability for system to recover.

The next challenge is about the inference. Obviously, Context Provision is designed to mitigate this challenge. It provides the required contexts to applications that are designed under various requirements and situations. Simple contexts can be retrieved from database directly, while the inference engine can extract high-level contexts. With additional information, well-designed user profiles, and a good learning mechanism, the accuracy of the analysis will be high enough for smart home applications.

The last challenge mentioned is the user-oriented design. The solution of this challenge may refer to Application Interface. Although it does not provide any user interface, the provided interface for smart home applications are well enough for these applications to design the user-oriented user interfaces.

To sum up, CASSHA overcomes the challenges of smart home applications with the functionalities of its four components: Context Processing, Context Representation, Context Provision, and Application Interface. Challenges of smart home applications are overcome by either a single component or by several components among four of them. These four components are well bound to each other and satisfy the requirements of smart home applications.

5 Conclusion and Future Work

In this paper, we have proposed CASSHA, a context-aware system for smart home applications, to overcome the challenges of smart home applications. CASSHA consists of four components, namely Context Processing, Context Representation, Context Provision, and Application Interface. With the functionalities of these components, the challenges for smart home applications can be overcome or mitigated.

However, there are still some other challenges for smart home applications. One of these challenges is the impacts of ethic, legal, and social implications. When smart home applications become popular in human's living, these new technologies will lead to some non-technical issues. The violation of privacy, the influence of living styles, the transformation of social structure, and so forth. These impacts may be hardly predicted, and can only be analyzed through a thorough research on behaviors of human beings.

References

1. Chen, H., Finin, T., Joshi, A.: An Ontology for Context-Aware Pervasive Computing Environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, volume 18, issue 3, pages 197-207, May 2004.
2. Choi, J., Shin, D., Shin, D.: Research and Implementation of the Context-Aware Middleware for Controlling Home Appliances. *IEEE Transactions on Consumer Electronics*, volume 51, issue 1, pages 301-306, February 2005.
3. Costa, P. D.: Towards a Services Platform for Context-Aware Applications. *Thesis for a Master of Science degree in Telematics from the University of Twente, Enschede, The Netherlands*, August 2003.
4. Covington, M. J., Fogla, P., Zhan, Z., Ahamad, M.: A Context-Aware Security Architecture for Emerging Applications. *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002)*, pages 249-258, December 2002.
5. Edwards, W. K., Grinter, R. E.: At Home with Ubiquitous Computing: Seven Challenges. *Proceedings of the Third International Conference on Ubiquitous Computing (UBICOMP 2001)*, pages 256-272, September 2001.
6. Efstratiou, C., Cheverst, K., Davies, N., Friday, A.: An Architecture for the Effective Support of Adaptive Context-Aware Applications. *Proceedings of the Second International Conference on Mobile Data Management (MDM'01)*, pages 15-26, January 2001.
7. Gu, T., Pung, H. K., Zhang, D. Q.: Toward an OSGi-Based Infrastructure for Context-Aware Applications. *IEEE Pervasive Computing*, volume 3, issue 4, pages 66-74, October-December 2004.
8. Hattori, M., Cho, K., Ohsuga, A., Isshiki, M., Honiden, S.: Context-Aware Agent Platform in Ubiquitous Environments and its Verification Tests. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, pages 547-552, March 2003.
9. Lee, S. H., Chung, T. C.: System Architecture for Context-Aware Home Application. *Proceedings of the Second Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04)*, pages 149-153, May 2004.
10. Moon, K. D., Lee, Y. H., Kim, C. K.: Context-Aware and Adaptive Universal Home Network Middleware for Pervasive Digital Home Environment. *IEEE Consumer Communications and Networking Conference 2004 (CCNC 2004)*, pages 721-723, January 2004.
11. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. *First International Workshop on Advanced Context Modeling, Reasoning, and Management (UbiComp 2004)*, September 2004.
12. UPnP^{TM} Device Architecture. *UPnP Forum*, June 2000.