Context-aware collaborative media services for ubiHome

Yoosoo Oh, Choonsung Shin, Woojin Jung, and Woontack Woo

Abstract— **Context-aware services in a smart home have been a research focus in Ubiquitous Computing for several years. Many services have been developed in the research community using various sensors and acting on different contexts. However collaborative services in a smart home still raise central issues that need to be solved. In this paper we propose the ubiTV, a context-aware collaborative media service which recommends services (or contents) according to the resident's preference by resolving conflicts among multiple residents in a smart home. The proposed ubiTV recognizes each resident's activities based on sensing information (profile, location, orientation, etc.). It reasons out an explicit intention by fusing multiple context inputs from a set of sensors. Also, it recommends the proper service or content to a resident by resolving conflicts occurred. Eventually, the ubiTV can help promote a harmonious family in ubiHome, a smart home environment.**

Index Terms— **Conflict Resolution, Context Integration, Location-aware, Service Recommendation, User-centric context**

I. Introduction

Ubiquitous Computing enables a shift in paradigm from technology-oriented to user-centered. In particular a smart home is one example in which technologies are applied for flexible communication among family members. Context-aware services in a smart home have been a research focus in Ubiquitous Computing for several years. Many services have been developed in the research community using various sensors and acting on different contexts [10].

Residents in a smart home can be provided with user-centric services by ubiComp-enabling technologies which can help promote a harmonious family. To strengthen family ties in a smart home, it is a good approach to consider sensing technology to recognize a resident's attention, and techniques to monitor a resident's activity from various sensors. Also, it is helpful to consider problems that occur while providing services to multiple residents.

Recently, several research activities on smart home services have been reported. EasyLiving (MS Research, since 1998) [1], AwareHome (GATECH, since 1999) [2], Adaptive House (Colorado Univ., since 1999) [3], House_n Project (MIT, since 2000) [4] are some of them. However, the existing research activities have some limitations. First, it is hard to manage multiple residents' attention simultaneously in a smart home. Second, they do not show the proper method which integrates context inputs from various kinds of sensors to predict a resident's explicit intention. Finally, they overlook resolving conflicts among multiple residents. Moreover, collaborative services in a smart home still have central issues that are not solved.

Therefore, we present the ubiTV which a context-aware collaborative media service for multiple residents in a smart home. For the implementation, we discuss Location Awareness as sensing, Context Integration as reasoning, and Conflict Resolution as recommendation. The proposed ubiTV recognizes each resident's location and orientation by using the location tracking sensor, and tries to grasp multiple residents' activities simultaneously. It infers their explicit intention based on their activities by integrating multiple context inputs from a set of sensors. Also, it recommends the proper services or contents to a resident by resolving conflicts that can be occurred.

Furthermore, it has the following features. The ubiTV provides user-centric personalized services to residents by integrating and managing contexts obtained from heterogeneous sensors. It provides various services (TV, DVD, Music, etc.) and gives information (Weather, Stocks, etc.) to residents. Additionally, it enhances communications among members of the family. It introduces an example of technology to strengthen family ties by exploiting various home services. Therefore, it will be helpful for enriching family-oriented home by applying context based collaboration among media services.

This paper is organized as follows: In Section Ⅱ, we describe ubiHome test-bed with the ubiTV. In Section Ⅲ, we explain context-aware technology in the context-aware collaborative media services. Especially, we discuss Location Awareness, Context Integration, and Conflict Resolution. We show the implementation and experimental results in Section Ⅳ. Finally, we present conclusion and future works in Section Ⅴ.

II. ubiHome test-bed

The ubiHome is a test-bed for applying ubiComp-enabling technologies to the home environment [5]. Various kinds of pervasive sensors and services have been embedded in ubiHome. Those embedded sensors and services form the foundation of the integrated smart home for multiple residents.

This works was supported by the UCN Project, the MIC 21st Century Frontier R&D Program in Korea, under CTRC at GIST.

The authors are with GIST U-VR Lab., Gwangju 500-712, S.Korea (email: {yoh,cshin,wjung,wwoo}@gist.ac.kr).

Context in ubiHome is defined by adding time context (context history) in addition to Schilit's definition. The context is represented as a form of 5W1H (Who, What, Where, When, How, and Why) by simplifying it [5][6]. Context modeling in ubiHome is based on ubi-UCAM 2.0 which is a unified context-aware application model in Ubiquitous Computing environment [7]. It is used to recognize a resident's state, behavior and surroundings.

Pervasive sensors in ubiHome detect nearby residents and environments for the resident. As shown in figure 1, various kinds of sensors such as Couch sensor, IR sensor, PDA, ubiTrack, RF tag, etc. are deployed in ubiHome [5][8]. Each sensor individually acts as a smart sensor with sensing, inherent processing, and networking abilities.

Services in ubiHome are provided as user-centric services to multiple residents. The c-MP (context-based Media Player), c-Mail checker (context-based eMail checker), TMCS (Tangible Media Control System), and cPost-it (Context-based Post-it) are some of them [5][8]. Especially, the ubiTV provides user-centric media services to harmonize residents based on the ubi-UCAM 2.0.

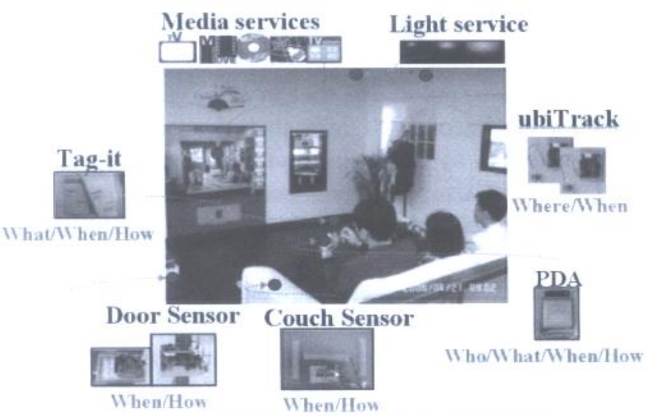

Fig.1. ubiHome test-bed with various sensors.

A. The ubiTV application

The proposed ubiTV is designed to make a harmonious family by exploiting various smart home services. To strengthen family ties, it technically supports to get the residents' experiences through the relationship among family members which consist of a married couple in the thirties and a young son. Thus, the ubiTV recommends the proper services or contents to the residents. The services-contents lists in the ubiTV are shown in Table 1. Additionally, it shows a guideline that expresses the expected experiences of residents in a smart home. Through this guideline, a developer can analyze the experiences of residents in a smart home in advance.

Table 1. A sort of Services/Contents in the ubiTV.

Services	Contents
Broadcasting	News, Education, Drama, Comedy, Animation
DVD	SF, Horror, Melodrama, Comic, Animation
Music	Rock, Ballad, Dance, Classic, jazz
Web	Game, Whether, Stocks, shopping, taxes

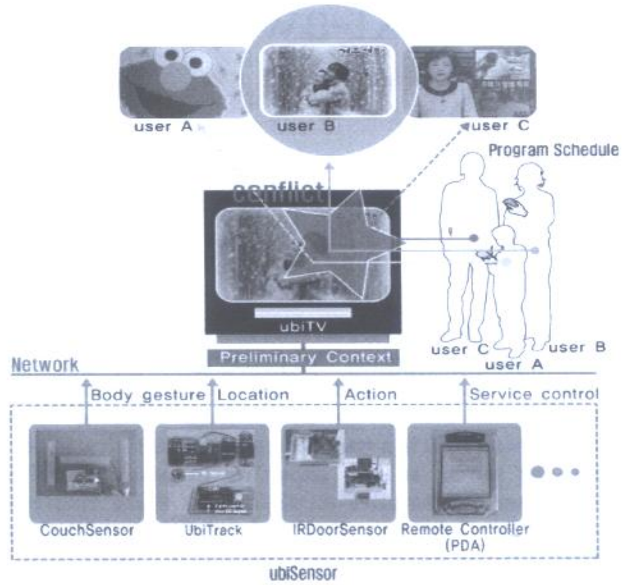

Fig.2. The implemented ubiTV.

The ubiTV has an architecture that recommends the proper services or contents in a menu for resolving conflicts. It also provides personalized services to multiple residents by exploiting ubi-UCAM 2.0. Figure 2 describes the implemented ubiTV. In figure 2, the ubiTV understands that User A, B, and C want to watch a TV at the same time by obtaining contexts from ubiSensors, and then it recommends the proper contents to the selected user by resolving conflicts. It means that the right of the choice is given to the selected user. But in real life, people can also take parts in decision by talking to each other.

The ubiTV can provide intelligent services according to a resident's explicit intention. That means it can predict his explicit intention. In real life, it's very hard to extract intention between human and human communications. Thus, the ubiTV makes decision by integrating the resident's activity to predict his explicit intention.

III. Context-aware Technology in ubiHome

Context-aware technology in the ubiTV is Location Awareness, Context Integration, and Conflict Resolution.

A. Describing Context: 5W1H

The representation of information, ranging from sensor reading to context and to actuators commands influences the design of context-aware systems. The 5W1H context is the structured format, where context is expressed as its components with regard to Who, What, Where, When, How, and Why. We used this representation to provide a resident centric view of context which is appropriate for the anticipated systems we want to support with the architecture. Using the 6 questions and structuring information gathered by sensors accordingly, allow us to accurately represent the resident's situation as it can be observed by sensors.

In addition this context representation can reduce the management and provision of context according for individual service. Each 5W1H context is hierarchically structured. *Who* context consists of a resident's identity, characteristics, and relationships. *Where* context comprises of a resident's indoor or outdoor location. *What* context is information of sensors and services being used by a resident. *When* context is time information generated by sensors or services. *How* context represents body conditions or gestures, or control command of a resident. And, *Why* context, which is abstracted from the concrete data that is gathered and combined, consists of a resident's expression, intention, and emotion. For a detailed description see [6].

B. Location-Awareness

The ubiTrack is the infrared-based location tracking system which tracks the location and orientation of residents and objects indoors [9]. In order to reduce the computational load, the ubiTrack uses a proximity method for tracking a resident's location. To lift accuracy of location tracking, the ubiTrack divides sensing area densely by using the overlapping area of signals from each transmitter. The ubiTrack is designed as user-centric system that maintains that all of the processing should be done in a mobile(personal) device and also controls the privacy as two levels. In addition, through two IR receivers attached on the shoulders, the ubiTrack can recognize the resident's orientation. The ubiTrack recognizes velocity and several states of the resident by using location and orientation information. By using the attachable ubiTrack which supports movable transmitters, it recognizes additional states of the resident, such as lying down.

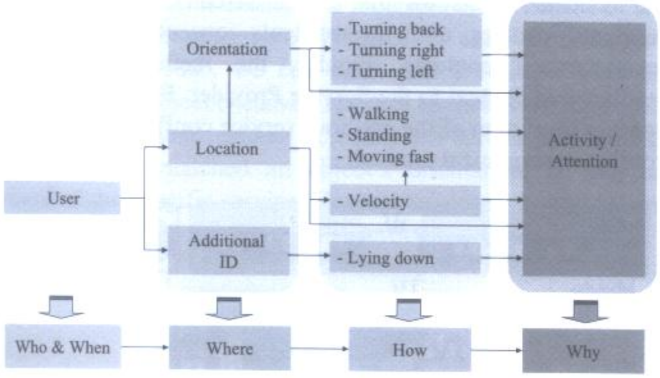

Fig.3. Data flow in the ubiTrack.

Figure 3 illustrates the data flow in the ubiTrack. Firstly, the mobile device gives the resident's ID, and it composes *who* information. Location and orientation are components of *where* information. Then, the ubiTrack generates *how* information which comprises of velocity and several states, such as standing, walking, and moving fast inferred from *where* information. The lying-down state, recognized from an additional ID is also one of the components of *how* information. Finally all of the information from the ubiTrack is transferred to ubi-UCAM 2.0 in order to infer a resident's activity and attention which are components of *why* information.

C. Context Integration

Context fusion and reasoning are the central functionality provided by the Context Integrator, which is a novel architecture for context-aware systems. An architectural overview is shown in figure 4. Context Integrator is composed of the Context Object Analyzer, the Context Repository, the Preliminary Context Fusion module, the Context Inference Engine, and the Integrated Context Generator.

The Context Object Analyzer collects contexts (preliminary or final alike) periodically from various kinds of sensors which are placed semantically in the same active area. The Context Repository stores and manages histories of the integrated contexts. The Preliminary Context Fusion module integrates the inputted Preliminary Contexts as an integrated 4W1H context according to characteristics of each sub-context of 4W1H (Who, What, Where, When, and How). Integrated Contexts are created by applying the appropriate fusion method. Preliminary Context (PC) expresses the feature information and sensor description. Integrated Context (IC) forms the complete 5W1H context generated by Context Integrator which additionally includes the inferred intention using the context inference engine. The Context Inference Engine reasons out *Why* component of the context (intention) by using the result of Preliminary Context Fusion module. It contains a knowledgebase which consists of facts and rules that describe the behavior of the system. The Integrated Context Generator combines all components into a 5W1H Integrated Context. This contains information, such as a resident's identity, location, activities, behavior, patterns, and intention.

Context Integrator is designed to provide information fusion on different levels and support fusion of sensor data from various sources. This makes results in a system that accomplishes symbolic context fusion. Fusion on raw data level is considered as a part of the sensor internals. By classifying and fusing according to a resident (*Who* dimension) as user-centric context integration is supported.

The Context Integrator has built in mechanisms for reuse of contexts in particular in the context fusion step. The reuse occurs in the inference chain which reuses the reasoned result of the previous step. With an increasing number of inference steps higher-level context can be created. This is the mechanism used to detect intention (*Why* component). Additionally, reuse is achieved by updating and using recorded contexts in the context history which are held in the Context Repository.

The generic behavior and functionality of a specific context-aware system running based on Context Integrator are specified by rules. Changes and reconfiguration of the system by a developer or maintainer are done by adding, deleting, and modifying rules. For simplicity these rules are specified in plain text and reconfiguration just requires a text editor. Hence changes to the overall system behavior do not require changes in any source code in Context Integrator. In the implementation of the Context Integrator, the Context Inference Engine is implemented on top of JESS (Java Expert System Shell). Rules are specified in syntax similar to LISP. Compared to declarative programming language this eases reconfiguration as rules that don't require order.

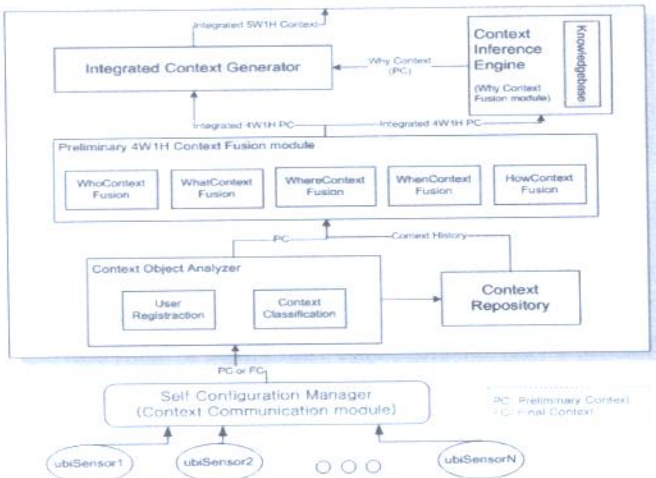

Fig.4. An architectural overview of the Context Integrator.

Context Integrator can infer a resident's behavior or gesture. At this point, previous contexts are important clues. Thus, context history is used to evaluate the resident's behavior. Context Integrator can get coordinates on both shoulders of a resident through the ubiTrack [9]. We assumed that left shoulder has the coordinate (x1, y1) and right shoulder has the coordinate (x2, y2). By using those coordinates, a resident's orientation can be calculated. Moreover, both coordinates and the orientation can be used to infer a resident's posture on a couch. If (x1, y1) and (x2, y2) are included in a sensor region, Context Integrator infers that the resident sits in a seat in the obtained direction. If (x1, y1) and (x2, y2) are included in different regions, Context Integrator infers that the resident occupies two or more seats in the direction. Thus, this inference can be used to predict a resident's attention.

D. Conflict Resolution

Conflict Manager is embedded in each service and provides conflict-free context as a Final Context to the registered service after resolving service conflicts [11]. First of all, the proposed method detects service conflicts by utilizing user contexts. The user context includes the resident's contextual information and the media service profile. Most of the context-based media services utilize such context information to provide residents with relevant information or services [7][12][13]. The proposed method intercepts the contexts delivered to the media service providers. It also exploits user contexts generated from other context-based media. Figure 5 shows the overall procedure of the conflict resolution.

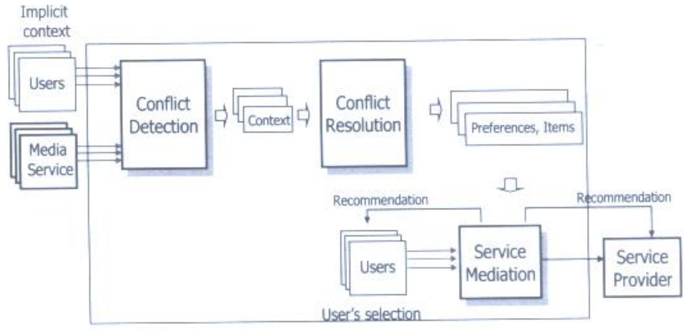

Fig.5. The procedure of conflict resolution.

As can be seen in figure 5, Conflict Manager gathers two types of user context: user context describing residents who are using the same media service and secondly user context describing residents who want to access different media services. In the conflict detection component service conflicts among residents accessing the same media service are detected by exploiting the resident and media service profiles. In the conflict resolution, the recommendation information is generated for a resident-centric conflict resolution by exploiting the context of the conflicting residents. The recommendation proceeds in two steps: The recommending of media contents regarding one media service and different media services. In the first step, the method obtains a recommendation list from the user contexts describing all residents who access the same media service by ordering media contents. It then determines recommendation precedence of media contents based on properties of the media services. In order to calculate the precedence, the proposed method assigns priorities to media contents, based on the utility of media services. Consequently, two kinds of recommendation lists are displayed for the service mediation after the conflict management process. Finally, the service mediation highlights the resident inputs on the shared screen to visualize the residents' choices, until the residents consent on an item of recommended contents. Finally, the method delivers a conflict-free context to the Service Provider. Figure 8 shows a conflict resolution of the previous service conflict based on the conflict management method.

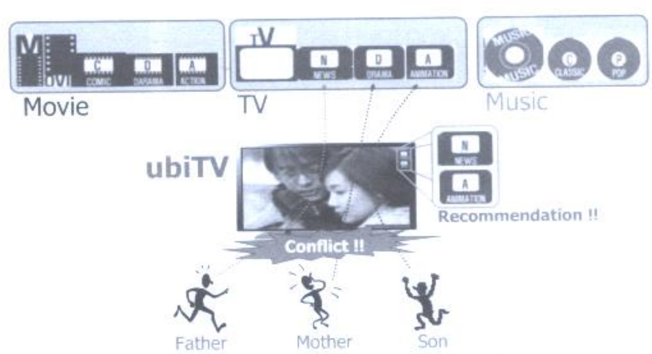

Fig.6. An example of the conflict resolution.

As can be seen in figure 6, a recommendation list of TV content is displayed on a shared screen and the resident's mobile devices when a service conflict is detected. According to family member's preferences, the recommendation list consisting of <drama, news, animation> reflects the preferences of father and son. The recommendation is also given to the residents and is reordered individually by each resident's preferences based on the resident's profile manager. Therefore, the father can see the recommendation consisting of news, sitcom and animation, and the son can see the recommendation consisting of animation, sitcom and news. If they choose an item of the recommended content, the selection is highlighted on the TV screen. Therefore, they can recognize each others preferences and opinions. They can also discuss about proper contents among the recommendation. Furthermore, if the mother, the third user, wants to access the TV service, her media service is similarly managed by the proposed conflict management method due to a service conflict with the TV service. Consequently a recommendation list, reflecting the mother's preferences is given to all conflicting residents. Therefore, the residents can choose an item of the media content to harmonize their preference at to solve the conflict in the shared space.

IV. Experiment

The ubiTV is implemented by using a physical TV device, a PC and several sensors, such as ubiTrack, Couch Sensor, IR DoorSensor, PDA, etc. in ubiHome. To evaluate the proposed components, we had several experiments.

To get the resident's activity clue, we use four information, such as location and orientation information of both a resident and an object. The ubiTrack is implemented on a resident and an object for location and orientation tracking of them. First of all, the ubiTrack of an object recognizes its location and orientation. Context Integrator makes the service area of an object. The ubiTrack of a resident recognizes his or her location. Context Integrator then find out whether a resident exists in a service area or not. If exists, then it compares orientation of the resident and the object. For instance, if each difference of orientations is 180°, we can think that a resident pays an attention to an object because a resident looks toward the object. In figure 7, because resident 'a' exists in a service area and his view point is matched an object's orientation, he can be provided the specific service. However resident 'b' and 'c' are cannot have a service about the specific object.

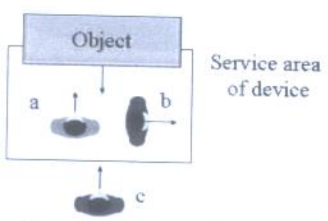

Fig.7. Residents' attention based their orientation.

In that view points, we did an experiment in order to find out the accuracy of orientation measured by the ubiTrack. We attached two IR receiver modules on the resident's shoulders, and measured his orientation by varying angles by 30° in the same position. To verify the reliability of the proposed orientation tracking method, we selected 10 points randomly, and in each point we measured orientation in 8 directions. The height of the receiver was 140cm and distance between two transmitters is 45cm, which is about the width of as average person's shoulder. Figure 8 shows experimental result. X axis is difference between real orientation and measured orientation. Y axis is the number of correct data measured by ubiTrack. We tested 80 orientations. The number of data having no error is 37 (46.25%). The probability that an error exists in -45 ~ + 45 degrees is 87.5. And the probability that an error exists in -90 ~ +90 degrees is 96.25%.

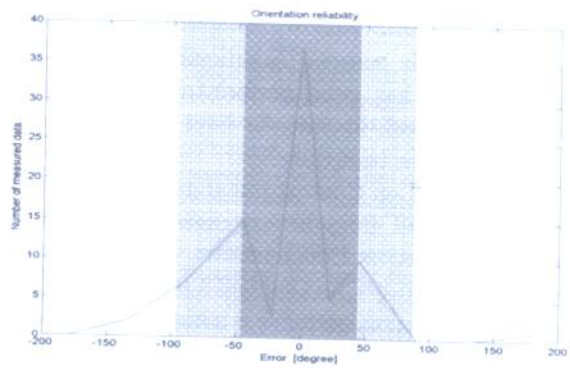

Fig.8. Orientation reliability.

There are two reasons that such errors are generated. Firstly, relatively small errors generated in -45 ~ +45 degrees are caused by using the proximity algorithm for location tracking. In that case, although the IR receiver is moved, the receiver still exists in the same sensing area of the IR transmitter, so that errors are generated. Secondly, relatively large errors are caused by the shape of the sensing area being not unique. Because of these two reasons, the ubiTrack has some errors in orientation tracking. In order to reduce them, each shape of sensing area should be uniform and smaller than now. If we change the shape of transmitters from current grid shape to triangle and decide optimized distance among transmitters, we can decrease the error.

Second, we measured performance on reasoning out a resident's explicit intention by integrating multiple context inputs obtained from various sensors in ubiHome. We experimented how Context Integrator performed by inspecting user-centric integration and inference from a set of sensors. Table 2 shows the performance of Context Integrator. Integration Interval means the time period that Context Integrator decides each integrated context. In this experiment, Context Integrator integrates 10 context inputs at once in a given Integration Interval. CPU occupying ratio represents the usage of CPU when Context Integrator integrates contexts. User-centric Integration (G/T) is a measurement of how *Who* context fusion affects the integration. It is a ratio between the number of the generated integrated context (G) with a resident's identity and total number of input (T) in a given interval. Its result expressing the resident's identity is important because *Who* context fusion classifies the context input by the resident's identity.

Table 2. The performance of Context Integrator.

Integration Interval	CPU occupying ratio	User-centric Integration (*G/T*)
0.1 sec	48 %	4/20
0.5 sec	32 %	1/20
3 sec	45 %	1/20

[PC Env.: CPU PIII 800M, RAM 512M, *G*: the number of the generated IC with user's identity, *T*: total number of input]

Accordingly, we could verify that Context Integrator is able to perform user-centric view by integrating multiple context inputs from multiple sensors. Additionally, we can see that Context Integrator has good performance when Integration Interval is 0.5 second, in order to infer a resident's explicit intention based on his behavior.

We experimented on Context Integrator with various types of contexts obtained from heterogeneous sensors. For this experiment, we used 3 CouchSensors, 3 ubiTracks, and 3 ubiControllers as sensors, and applied 3 residents as a target of the experiment. The ubiController is a kind of user interface to control services of ubiHome based on UPnP service discovery. It is implemented on a PDA-phone, as shown in figure 9. The CouchSensor is embedded in a couch, and delivered its center coordinates to discovered services in ubiHome. The ubiTrack also delivered the center coordinates of a resident's shoulders. Residents always carry with their ubiTrack and ubiController in ubiHome. Figure 10 shows x-y coordinates of each object. Preliminary context is represented by 4W1H context form, such as [Who What Where When How].

Fig.9. ubiController.

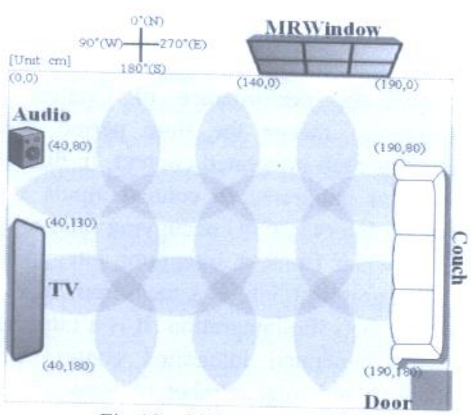

Fig.10. ubiHome coordinates.

As our experiment, we could know there are 3 persons in ubiHome, and expect to be created 3 integrated contexts. In fact, we could get 3 integrated contexts in a given interval, 0.5 second. Those are [Father TV West of Couch 20050830171130 SitDown ToWatch(News)], [Mother Unconcern West of Couch 20050830171130 StandUp Normal], [Son WindowDisplay South of WindowDisplay 20050830171130 Selection ToSee(Family picture)]. This result represents that Father sits on a couch and wants to watch news on TV, and Mother has no attention at this time. It also shows that Son pays attention to WindowDisplay, and wants to see a family picture. At this point, news and a family picture are obtained from residents' preferences in their profile. As the result, we could know that the proposed context integration method is efficient to seize the proper service for each resident.

In the following experiment we wanted to get a first impression on how residents react on recommendation. In order to do that, we surveyed the behavior of 12 users, concerning the use of media services in a smart home environment. Firstly, we took a closer look at the service recommendation for conflict resolution. The goal of the recommendation list is, besides supporting a discussion, to help the users to make a fast and convenient decision. The whole process is supported by visualizing each family member's preferences.

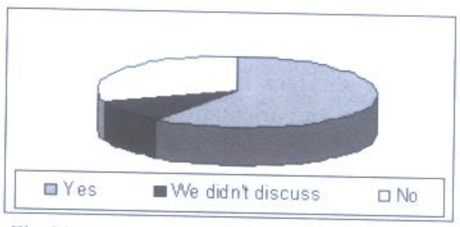

Fig.11. Recommendation helping discussion.

As can be seen figure 11, sixty percent of the asked users answered that they felt supported by the recommendation list in the discussion process. This clearly shows that the visualization of other people interests can support a verbal discussion.

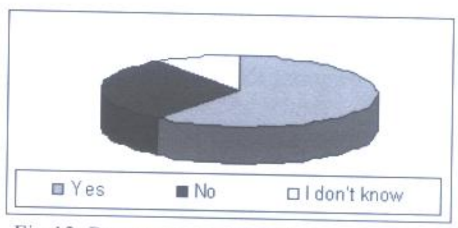

Fig.12. Recommendation making a decision.

As shown in figure 12, the analysis of the results show that 60% of the experiment participants felt that the recommendation list supports the decision making process. Based on these results, it seems that the recommendation list is a proper technique to support the whole process to harmoniously choose the TV content in a family environment.

In order to measure the usefulness of the resolution method of the proposed Conflict Manager, we experimented on user conflict in such setting. To test the method, we employed a television service that users spend most of their time on watching in their home. While using the television service, family members cause conflicts due to their preferences and its broadcasts. In our experiment, the television service recommends preferred genres of conflicting users in that conflict situation. The service then gathered feedback of users in pre-defined amount of time and judged the hit on the recommendation. We did this experiment from 18:00 to 24:00 in two weeks and obtained 185 conflicting samples from three users. Finally, we have built a hit matrix to know how well it works. Table 3 shows hit ratio on the recommended genres of each users.

Table 3. Hit ratio on recommended genres (unit :%).

Users	Hits	News	Drama	Education	Animation	Etc
Father	38	34	10	10	5	41
Mother	41	17	27	43	4	9
Son	44	12	21	17	22	28

As shown in Table 3, the users show different characteristics on the recommendation. Father expressed higher selection on News program than that of other programs, but showed relative lower hit ration than other users. In case of mother, most of the hits are related to Drama program. Son selected various programs compared with father and mother. He also shows relative higher then the users. However, they were unlikely to choose their preferred program soon after content recommendation. This is because Conflict Manager enabled them to spend their time on talking about the current program with recommended programs. They then decided a suitable program to accommodate those users. Especially, mother encourages her son to watch educational programs when they were in together.

V. Conclusion

In this paper, we presented the ubiTV which can help promote a harmonious family in ubiHome. This paper prepares a chance that enriches their mentality by providing various services of the ubiTV to multiple residents in a smart home. It establishes a foundation for user-centric personalized services based on residents' attention or intention. Additionally, the proposed ubiTV shows an application of technology to strengthen family ties by exploiting various home services. In near future, we will experiment the situation for resolving conflicts among family members in various daily activities and will do the usability test to evaluate whether our recommendation method is proper for every involved residents.

References

[1] S.Shafer, B.Brumitt, and B.Meyers, The EasyLiving Intelligent Environment System, CHI Workshop on Research Directions in Situated Computing, 2000.

[2] Sensing the Subtleties of Everyday Life, the research magazine of Georgia Tech. (Research Horizons), 2000.

[3] M.C.Mozer, An intelligent environment must be adaptive, IEEE Intelligent Systems and their Applications, 14(2), pp. 11-13, 1999.

[4] House_n Research Group, http://architecture.mit.edu/house_n/, 2005.

[5] S.Jang, C.Shin, Y.Oh, and W.Woo, Introduction of "UbiHome" Testbed, The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems 2005(ubiCNS2005), 2005.

[6] S.Jang and W.Woo, 5W1H: Unified User-Centric Context, The 7th International Conference on Ubiquitous Computing(UbiComp05), pp. 2005.

[7] Y.Oh, C.Shin, S.Jang, and W.Woo, ubi-UCAM 2.0: A Unified Context-aware Application Model for Ubiquitous Computing Environments, The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems 2005(ubiCNS2005), 2005.

[8] Y.Oh and W.Woo, A unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness, UCS2004, pp. 117-122, 2004.

[9] W.Jung and W.Woo, Indoor orientation tracking using ubiTrack, The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems 2005(ubiCNS2005), 2005.

[10] H. Wu. Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory, PhD Thesis, Carnegie Mellon University, 2003.

[11] C.Shin and W.Woo, Conflict Resolution Method using Context History for Context-aware Applications, Workshop of Pervasive05, 2005.

[12] Anind K. Dey and Gregory D. Abowd, The Context Toolkit: Aiding the Development of Context-Aware Applications, *Proceedings of the Workshop on Software Engineering for Wearable and Pervasive Computing (SEWPC)*, Limerick, Ireland, June 6, 2000.

[13] Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces.", *In IEEE Pervasive Computing*, pp. 74-83, Oct-Dec 2002.