Conflict Management for Media Services by exploiting Service Profile and User Preference

Choonsung Shin, Daeho Han and Woontack Woo

***Abstract*— In this paper, we propose Context Manager to resolve conflicts for Context-aware Media services in smart home environments. Conflicts arise when multiple users access a Media service or when various Media services share limited resources to provide customized responses. In order to resolve conflicts among users, the Context Manager sums up preferences of users who collide with each other and recommends specific contents ordered by the summed preference. It also resolves conflicts among Media services by selecting a Media service with the highest preference. Furthermore, Context Manager resolves conflicts among Media services occupied by users with the Media service recommendation. During experiments on ubiHome, a smart home test-bed, the proposed method resolved conflicts among users and Media services while giving higher satisfaction to users than a resolution method selecting a specific user. Therefore, we expect the proposed management method can play a vital role in Context-aware Media services for offering personalized services to users by resolving service conflicts among Media services as well as users.**

***Index Terms*—Context-awareness, Service Conflicts, Service Profile, Media Service, User Preference**

I. Introduction

THE aim of ubiquitous computing is to provide users with intelligent services based on the information obtained from distributed and invisible computing resources [1]. Users exploit these services without any cumbersome interface or learning procedures. Therefore, context-aware applications are considered to be important applications in the ubiquitous computing environments that offer appropriate services to users by utilizing contextual information of users and their surroundings [2][3]. Especially, context-aware Media services are one of the emerging applications for smart home environments. The Home Media Space (HMS) allows users to teleconference with remote users through video streaming [4]. Context-based Media Player (cMP) offers various movie contents by utilizing user's contextual information and preference [6]. Music Player Application provides functions for playing music by exploiting distributed resources, such as display and audio devices in Active Space [5].

However, conflicts occur in context-aware media services when more than one user reaches the Media service simultaneously due to sharing of the space and service. Hughes pointed out that conflicts arise when residents share the appliances, such as radio, television, and etc. [7]. W.Keith classified the conflict problem among users as a challenge in ubiquitous computing home environments [8]. Besides, conflicts also arise when more then one Media service shares the limited resources within their environment. Ajay A considered this kind of conflicts as conflicting reactions in behavioral system [13]. Conflicts also arise when more than one user access different Media services. Meyer defined the ubiquitous computing environment as multi-user and multi device environments [9]. Furthermore, S.M. Easterbrook pointed out the complexity of home environments like "the less cohesive the group, the more conflicts" [11]. Consequently, in order to develop Context-aware Media services for home environments, it is necessary to resolve conflicts not only among Media services and among users, but also among Media services provided to different users.

Meanwhile, most research, aimed on resolving conflicts, has been done on smart home and intelligent office. MusicFX supports group music selection in a fitness center by exploiting users' preferences on each music station [12]. Reactive Behavioral System (ReBa) resolves conflicts among devices in office environments by applying layering architecture of activity bundles consisting of users' activities and reactions of environments [14]. Reconfigurable Context-Sensitive Middleware for Pervasive Computing (RCSM), an object-based framework, ensures independence between sensors and application services, forms ad-hoc communication between them, and delivers the necessary context to the applications [16]. Context Toolkit collects, interprets, and delivers context between sensors and application services [15]. Contextual Information Service (CIS) manages contextual information such as location and characteristics of users, devices, and status of network to provide contexts to application services [17].

Nevertheless, context management techniques in the previous research have various limitations when they are applied to multi-user environments with various applications. MusicFX only utilizes users' preferences on all situations of users and lacks consideration of multiple services [12]. In case of ReBa, it is difficult to provide particular services to each user because ReBa focuses on the service for grouped users by inferring main activities from the environment [14]. In the RCSM, context management does not consider shared devices

Manuscript created June 17, 2005. This work was supported by Samsung Electronics Co., Ltd. in S.Korea

Choonsung Shin is with Gwangju Institute of Science and Technology, Gwangju, 500-712, S.Korea (e-mail: cshin@gist.ac.kr)

Deaho Han is with Samsung Electronics Co., Ltd, Seoul, 135-284, S.Korea (e-mail: bighoya.han@samsung.com)

Woontack Woo is with Gwangju Institute of Science and Technology, Gwanju, 500-712, S.Korea (Corresponding author to provide phone: 82-62-970-2226, fax: 82-62-970-2249, email: wwoo@gist.ac.kr)

or services because contextual information services are provided only through individual device possessed by each user [16]. In the case of Context toolkit and CIS, application developers have to consider both conflicts among services and among users since contexts are delivered to applications when current context of environment matches an application-specified condition [15][17].

In order to resolve the above-mentioned limitations and conflict problems, we propose Context Manager. It consists of four components: Context Preprocessor, Service Profile Manager, Final Context Deliverer, and Conflict Manager. Context Preprocessor matches and filters contexts coming from users and services and provides them to Service Profile Manager. Service Profile Manager maintains static and dynamic information of a registered Media service. With Profile Manager and Context Preprocessor, Conflict Manager detects and resolves the conflicts in context-aware Media services. Final Context Deliverer then sends the conflict-free context to the registered service and other Media services. Therefore, the proposed Context Manager resolves the conflicts among users by recommending specific contents sorted by the preferences of conflicting users. It also detects conflicts among Media services by utilizing their service profile and resolves by selecting a Media service with the highest preference. Furthermore, it resolves conflicts among Media services involved with different users by recommending the services sorted by the preference of the conflicting users.

Through implementation and experiment, we found that the proposed Context Manager has the following advantages. First, the recommendation of services and contents gives users higher satisfaction than the resolution method that selects a user or service when service conflicts arise. In addition, with recommendation, users become aware of others' needs as well as preferences. Therefore, the proposed resolution method can play an important role to congregate family members while resolving conflicts among them and Media services.

This paper is structured as follows. In Chapter 2, we introduce context-aware application model for ubiquitous computing environments and context-aware Media services. Chapter 3 describes the architecture of Context Manager and the conflict resolution method. Chapter 4 describes how the Context Manager manipulates service profile information. Experimental setup and results are discussed in Chapter 5. Finally, we conclude in Chapter 6.

II. Context-aware Media Service

A. Context-aware Application Model

In order to provide the personalized service, we utilize unified context and ubi-UCAM 2.0. The unified context represent a user's context information as 5W1H (Who, What, Where, When, How, and Why) [19]. 5W1H contains comprehensive information about user and his surrounding environment. With each field having sub-fields, the unified context also represents detailed information about the user. The unified context expressed with 5W1H ensures the independence between sensors and services. It also has the advantage to be re-used by other services. In addition, the unified-context ensures reducing additional management required to translate the context into other forms according to individual services. Unified context employs the different types of contexts according to the role of each context. These include Preliminary Context, Integrated Context, User Conditional Context, Service Conditional Context, and Final Context.

Preliminary Context (PC): A unified context which describes current situation of a user and his environment, and includes all or part of 5W1H.
Integrated Context (IC): A unified context which describes current situation of a user and his environment, and includes all of the 5W1H.
Conditional Context (CC): A unified context which expresses an action and parameters of a service and related user condition.
Final Context (FC): A unified context which describes a user, his environments, and service action and parameters. It is used for triggering a service.

ubi-UCAM 2.0 (Unified Context-aware Application Model for ubiquitous computing environment) is context-based application model to provide users with the personalized service by exploiting context in ubiquitous computing environments where various kinds of sensors and services are distributed [20]. Ubi-UCAM 2.0 is composed of ubiSensors and ubiServices. The ubiSensors and ubiServices exchange contextual information with several types of contexts based on the unified context. Figure 1 shows the overall architecture of the ubi-UCAM 2.0.

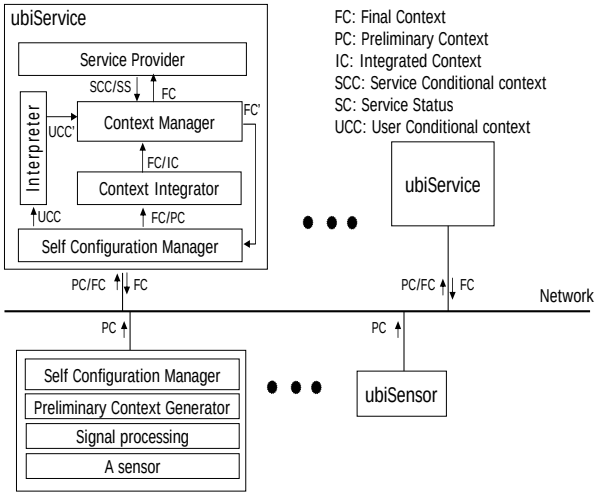


Fig. 1. Architecture of ubi-UCAM 2.0

As shown in Figure 1, ubi-UCAM 2.0 consists of ubiSensors and ubiService. A ubiSensor is composed of a physical sensor, Signal Processing module, Preliminary Context Generation module and Self Configuration Manager. The physical sensor perceives a change related to a user and his environment. Signal processing module extracts feature information from the sensed signal. Preliminary Context Generation module generates a preliminary context from the feature information. The

ubiSensor delivers the context to ubiServices located within a working area through a multicast group established by Self Configuration Manager. A ubiService is composed of Context Integrator, Context Manager, Interpreter and Service Provider. Context Integrator collects preliminary contexts created by various ubiSensors located within a working area during a given time interval. It classifies the preliminary contexts to each sub-element and analyzes the sub-elements by applying a decision making technique. Context Integrator generates integrated contexts of each user and delivers them to Context Manager. Context Manager searches conditional context from a Hash-table, which manages specific service action and condition, corresponding to each integrated context. It generates a final context to be used by applications after resolving conflicts among users and services. Finally, Service Provider executes an appropriate action with the parameters described in the final context.

B. *Context-aware Media Services*

ubiTV is a set of context-aware application that provides users with various Media services in smart home environments [20]. ubiTV is based on ubi-UCAM 2.0 and offers various media services, such as television, music, movie, and Internet service, according to users' contextual information. Figure 2 shows the overall architecture of ubiTV application.

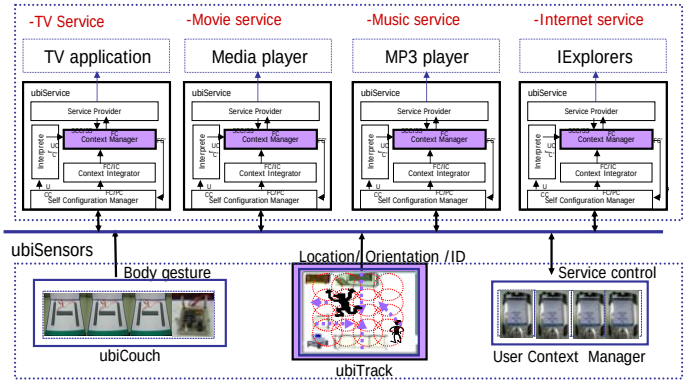


Fig. 2. The architecture of ubiTV

As shown in Figure 2, the ubiTV application is composed of Media services, a kind of ubiService, and various sensors, a kind of ubiSensor. Television service consists of Samsung PVR ™ application and ubiService. It provides users with various programs through 4 broadcasting channels in Korea: Korea Broadcasting System (KBS), Seoul Broadcasting System (SBS), Munhwa Broadcasting Corporation (MBC), and Educational Broadcasting System (EBS). Movie service is composed of Media Player ™ and ubiService. It provides four genres of movies, such as, Sci-Fi, Korean, Animation, etc. Music service consists of MP3 player and ubiService. It offers several genres of music stored in the computer. Lastly, Internet service, comprised of Internet Explorer ™ and ubiService, offers useful web pages such as weather, stock, and game. These services are deployed on the same computer having a large display and a surround speaker in the smart home test-bed. In addition, ubiTV also exploits various sensors: ubiCouch sensors, ubiTrack. The ubiCouch sensors are couch sensors comprised of on/off switches and PIC16F84, to detect user's behaviors. ubiTrack is infrared-based location tracking system that tracks user's location [21]. Furthermore, in order to control these services, users utilize ubiRemocons, a remote controller, implemented with Personal Java [29].

Based on the Media services and sensors, ubiTV application provides residents with the Media services according to users and their environments.

C. *Service Conflicts in Context-aware Media Services*

Unlike single user and single service environment, applications in the ubiquitous computing environment have to respond to users while considering other services and applications, along with various users within a services area. In case of ubiTV, it provides users with various Media service by exploiting users' contextual information within a service area. In order to provide services, the ubiTV application requires one or more resources, such as display device, sound device, or both, according to their properties. Furthermore, in such service environments, the number of users accessing the same Media service is not limited. Therefore, the Media services all the time cannot provide users with their personalized responses

We define such situation as a service conflict [13]. In the case of ubiTV, we classify service conflicts into three types according to sources of conflicts: service conflicts among multiple users, service conflicts among multiple Media services and service conflicts among multiple users and multiple Media services. Service conflicts among users are caused by use of a Media service by multiple users. In this situation, involving several users, the Media service has to choose one user who will be offered with the customized service. Figure 3 illustrates a service conflicts scenario among users

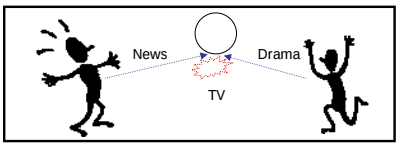


Fig. 3. A service conflict among users

As shown in Figure 3, a service conflict arises when users A and B are trying to watch their preferred broadcasts from television service. Therefore, television service has to choose one user. Next, service conflicts are caused by providing services among multiple Media services. Due to the conflict, the Media service cannot provide users with customized responses. Figure 4 shows a conflict among Media services.

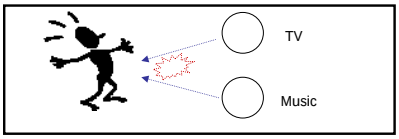


Fig. 4. A service conflict among Media services

As can be seen in Figure 4, television application and music application start to provide their customized services

simultaneously. Therefore, their sound interferes with each other. Service conflicts among users and applications are caused by the use of multiple services by multiple users. This kind of conflict scenario is similar to the conflict among multiple Media service, but different users are assigned to each Media service. Figure 5 shows a service conflict scenario among users and Media services.

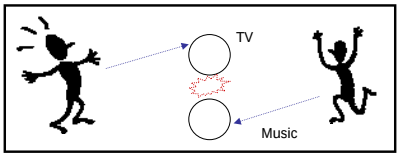


Fig. 5. A service conflict among users and Media services

As shown in Figure 5, a service conflict arises when user A is trying to use a television application while user B is trying to use a music application. Therefore, both television and Music can not provide customized responses to each user.

To deal with these conflicts, resolution methods have to resolve the conflict according to sources of conflicts. Furthermore, in order to reflect the change of users' preferences and their environment, the conflict resolution methods must adapt to users and their environment. In this paper, we deal with these three kinds of conflicts: among users, among Media services, and among Media services given to different users.

III. CONFLICT MANAGEMENT

In ubiTV application, Context Manager is located in each Media service and provides Final Context to the registered service. In order to provide Final context, Context Manager manages conflicts in its point of view. In case of conflicts among users, the Context Manager resolves them by recommending contents of conflicting users. It also plays a role in resolving conflicts among Media services by utilizing Final Contexts obtained from the Media services and service profile of a registered service. Furthermore, in order to resolve conflicts among Media services given to different users, Context Manager groups the conflicting services by each user and then recommends the services to users. Figure 6 shows the overall architecture of Context Manager.

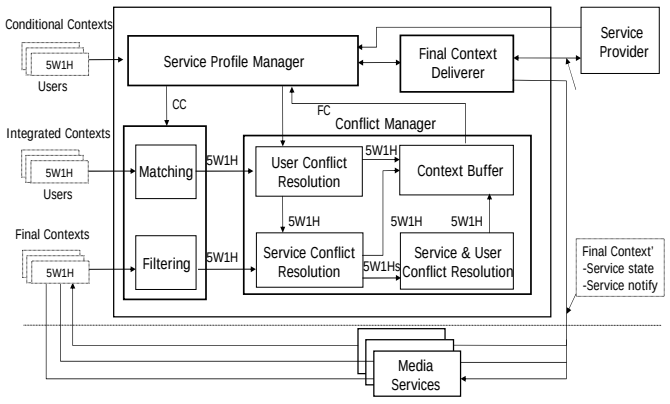


Fig. 6. The architecture of Context Manager

As can be seen in Figure 6, first of all, Context Manager receives three types of contexts, user context, a kind of FC, service contexts, a kind of FC, and User Conditional Context, a kind of CC. It then resolves conflicts among users with the user contexts. After resolving conflicts among users, Context Manager builds final context based on the conflict-free context. At that time, it tries to resolve conflicts among Media services. In case of single user, it selects a media service with the highest preference among users' Media services. Context Manager then sends the selected Media service to Context Buffer. In case of multiple users, it selects a Media service of each user's Media service .Conflict Manager then resolves conflicts among users and Media services by making a final context which implies service recommendation. Finally, Context Manager delivers the final context to Service Provider and other Media services within the same service area.

A. Conflict Resolution among Multiple Users

Conflict Manager resolves conflicts caused by users trying to use services within the same service area. To resolve the conflict, Conflict Manager manipulates user contexts in two steps: building a user conflict list and recommending preferred contents from the list. Figure 7 illustrates the resolution procedure on service conflicts among users.

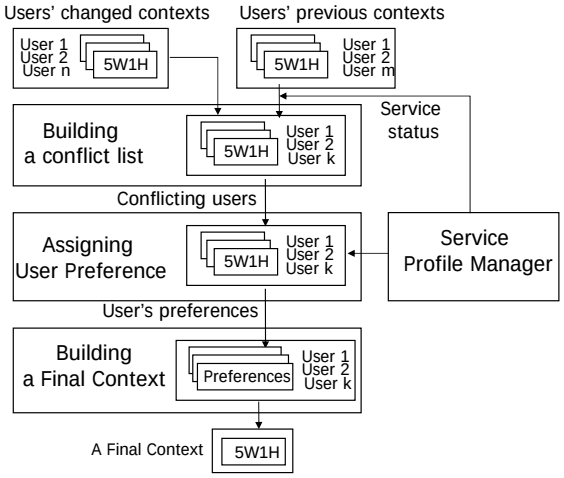


Fig. 7. Conflict Resolution among Users

As can be seen in Figure 7, first of all, Conflict Manager makes a conflict list of matched user context on users who are expected to cause conflict among users, including those who are currently using the services. In this procedure, users who leave the service area are excluded from the list because we assume they do not want to enjoy the service any longer. It then assigns preferences obtained from Service Profile Manager to each user's context. Consequently, Conflict Manager builds each User Preference (UP_{ij}) represented by Equation (1), where j is each preference of contents, i is each user, and k is the number of contents of the registered service.

$$User\ i\text{'s preferences} = \{UP_{i1}, UP_{i2}, \bullet\bullet\bullet, UP_{ik}\} \quad (1)$$

Conflict Manager then builds a Final Context having preferred contents ordered by users' preferences. In this process, conflicts are handled in several ways according to the number of users within the service area. In the case of one user, we know that there is no conflict among users. Therefore, Conflict Manager just selects the user context as a result of conflict resolution. However, we have to consider the situation when there is more than one user within a service area. In this situation, Conflict Manager sums up each user's preference on contents. In order to obtain the summed preference, we utilize McCarthy's Group Preference (GP_j) algorithm defined by Equation (2), where N is the number of users, i is each user, and j is the number of contents [12].

$$GP_j = \sum_{i=0}^{N} UP_{ij} \quad (2)$$

Conflict Manager then builds a final context with the summed preferences GP_j. Finally, Conflict Manager delivers the Final Context to Context Buffer. In case of user selection, Conflict Manager also offers the final context to service conflict resolution module. Figure 8 shows a service conflict scenario among multiple users.

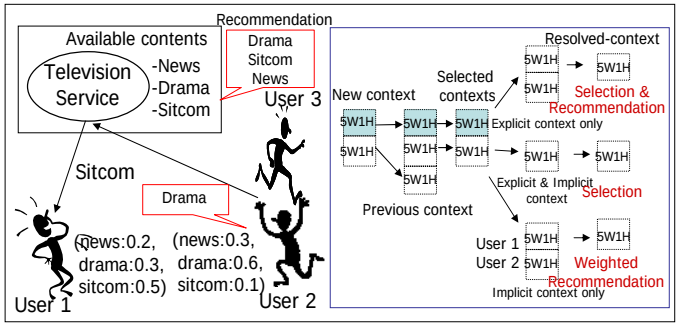


Fig. 8. A conflict scenario among multiple users

As shown in Figure 8, there is a television service providing user 1 with a sitcom program in a service area. Simultaneously, user 2 is trying to use the same service and user 3 is leaving from service area. Therefore, a service conflict arises due to use of the same television services by two users. In this conflict situation, Conflict Manager builds a conflict list consisting of contexts of user 1 and user 2, depicted in Table 3.

Table 3. A conflicting user list

	Who	What
FC 1	User 1, Preferences = {News: 0.2, Drama:0.3, Sitcom, 0.5}	Television, News
FC 2	User 2, Preferences = {News: 0.3, Drama:0.6, Sitcom, 0.1}	Television, Drama

Based on the group preference algorithm, Conflict Manager sums every preference of each user based on the conflicting list. In the scenario, user 1 has the ordered preference on contents of television service: 0.2, 0.3, and 0.5 to News, Drama, and Sitcom, respectively. User 2 has the ordered preference on contents of television: 0.3, 0.6, and 0.1 to News, Drama, and Sitcom, respectively. Summarizing the preferences, Conflict Manager obtains the preference of 0.5, 0.9 and 0.6 on the contents of the service. Consequently, Conflict Manager obtains a content list containing Drama, Sitcom and News programs ordered by the preference value.

B. Conflict Resolution among Multiple Media Services

Conflict Manager also detects and resolves conflicts caused by multiple Media services that share limited resource in a service area. In order to detect possible conflicts among media services, Conflict Manager manipulate contexts in two steps: building a conflicting service list and building Final Contexts. Figure 9 shows a conflict resolution procedure on service conflicts among Media services.

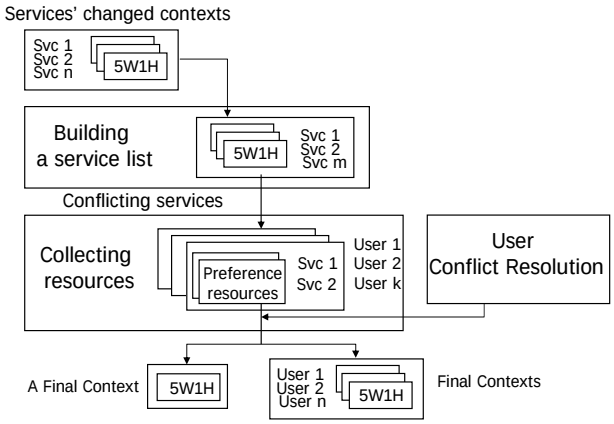


Fig. 9. Conflict Resolution among Services

As shown in Figure 9, Conflict Manager collects FCs of other services and monitors resource usage of other services within the same service area. Conflict Manger then builds a FC by exploiting the collected services and a FC of the registered Media service. Figure 10 shows a conflict scenario among Media services and a resolution procedure on the scenario.

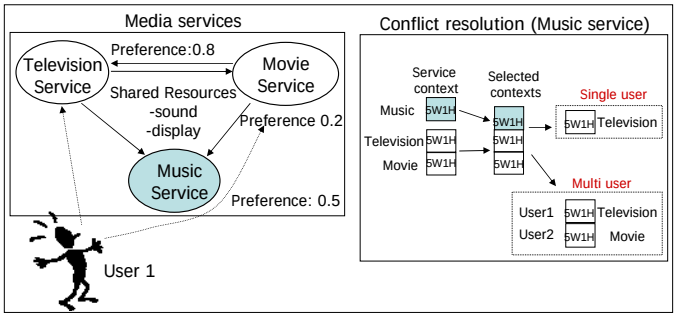


Fig. 10. A conflict scenario among Media Services

In the service scenario, Music service, Television, and Movie service share a sound resource of Television. Furthermore, Television and Movie services need a display resource to show their visual contents. Music service is providing user 1 with Music service. At that time, Television and Movie services are about to start. Therefore, due to the sharing of a sound resource, Television and Movie service cause a conflict with Music service. Conflict Manager of

Television and Movie services send a FC containing their status to Music service. Music service than receives those FCs and builds a conflict service list. The list also includes a user's context coming from user conflict resolution module. Table 4 shows the conflicting service list.

Table 4. A conflicting service list

	Who	What
Music	User 1	Music, On, Resources ={sound}, Preference = 0.5
FC 1	User 1	Television, On, Resources = {sound, display}, Preference = 0.8
FC 2	User 1	Movie, On, Resources = {sound, display}, Preference = 0.2

As can be seen in Table 4, Conflict Manager of Music service recognizes the sound conflict with Television and Movie service. It then gathers conflicting resources which are occupied by other Media services with higher preference than its registered service. According to the preference, Conflict Manager includes resources of the Television service, but excludes resources of Movie service. Finally, it delivers the resource list containing a sound element as a resolution result to Context Buffer.

C. Conflict Resolution among Users and Media services

Lastly, Conflict Manager resolves service conflicts caused by multiple Media services occupied by different users. In order to resolve expected service conflicts, it utilizes user's services list coming from service conflict resolution module. Figure 11 shows the resolution procedure on service conflicts among Media services and users

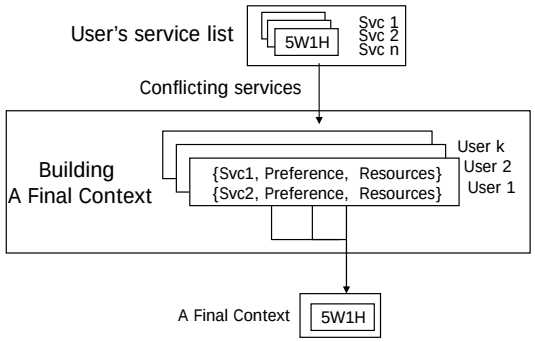


Fig. 11. Conflict Resolution among Media services and Users

As shown in Figure 11, Conflict Manager builds a conflicting service list from FCs. It then gathers the conflict service name to be used for the recommendation ordered by service's preference. Conflict Manager then builds a recommend service list by gathering names of services. Figure 12 shows a service conflict scenario among Media service given to different users and a resolution procedure on the scenario.

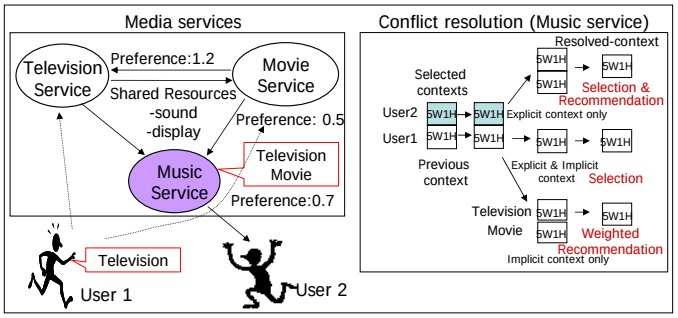


Fig. 12. A service conflict among users and Media services

As can be seen in Figure 12, the scenario is similar to the conflict scenario in multiple Media services. However, the users involved in each service are different. Music service has been occupied by User 2 and Television service is about to be engaged by User 1. Therefore, Conflict Manager of Music service builds a conflicting list containing Music, Movie and Television services, depicted in Table 5.

Table 5. A conflicting service list

	Who	What
Music	User 2	Music, On, Resources = {sound}, Preference = 0.7
FC 1	User 1	Television, On, Resources = { sound, display}, Preference = 1.2
FC 2	User 1	Movie, On, Resources = {sound, display}, Preference = 0.5

Conflict Manager then classifies the services to each user group according to the user of each service. Based on the list, it selects each representative service from user's services. Therefore, Music service and Television services are assigned to user 2 and User 1 respectively. Table 6 shows the selected services.

Table 6. A representative service list

	Who	What
FC 1	User 2	Music, On, Resources ={sound}, Preference =0.7
FC 2	User 1	Television, On, Resources = { sound, display}, Preference = 1.2

Finally, Conflict Manager builds a Final Context consisting of conflicting services based on the service list. In this case, the list is comprised of Music and Television services ordered by users' preference. Therefore, Conflict Manager obtains a Final Context having Music service and Television services.

IV. SERVICE PROFILE MANAGEMENT

As mentioned before, Context Manager has a role in resolving conflicts related to a registered Media service. Therefore, it requires static and dynamic information of the service. Service Profile Manager has a role in managing the profile information.

The next session explains how Service Profile Manager maintains the service profile information.

A. Service Profile

Service Profile is the information which is used to resolve service conflicts. It combines wide range of information about the service: Service Information, Conditional Contexts, Service Status, etc.

1) Service Information

Service Information is unique information used for identification within a service area. It consists of Service Name, Resources, and a Service Area. Service Name is a unique name and used to identify it within a service area. The service name enables Context Manager to identify each Media service within a service area. Resources are the devices needed for Media service to provide services. A Service Area is a location of the registered service. It also implies the area of service conflicts among applications

2) Conditional Contexts

Conditional Conditions are conditions of an application to provide users with customized services. They are divided into two types of conditions: User Conditional Contexts (UCC) and Service Conditional Contexts (SCC). The UCCs are obtained from User Profile Manager (UPM), a kind of wearService [27][28]. On the other hand, SCCs come from Service Provider and are configured by application developers.

3) Service Status:

Service Status describes currents status of the registered Media service including user's context and is represented with 5W1H. "Who" is a user who currently uses the Media service. "What" is a current state of the Media service. The Media service can be stayed on a state among service states: On, Off, Waiting, Pause and Recommending. These states are used to recognize service conflictions. "When" maintains time information when the user started to use the registered service. "Where" is a user's location within a service area. "How" is user's current state. "Why" is the reason why a current user wants to spend his time on the service.

B. Service Profile Manager

In order to manage the profile information of the registered service, Service Profile Manager (SPM) obtains service information from Service Provider and UPM, and provides the information to Conflict Manager and Context Deliverer. Figure 13 shows the overall architecture of SPM

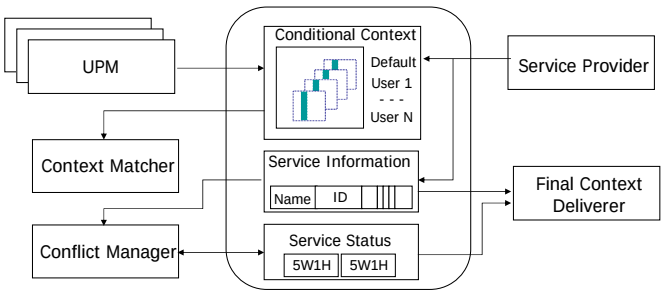


Fig. 13. The architecture of Service Profile Manager

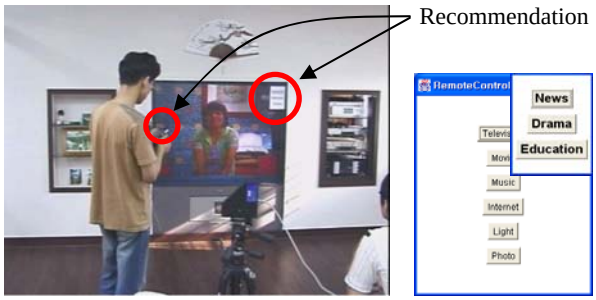
As can be seen in Figure 13, SPM requests properties of the registered service from Service Provider when an application starts serving users. The requested information is the Service Information of the registered service. SPM also receives SCC from the Service Provider. Furthermore, SPM collects UCCs from UPMs of wearServices of users' personal stations such as PDA while serving users. SPM manages the UCCs with Hash-table and provides them with Context Matcher to match ICs. SPM also offers Service Status information to Conflict Manager and receives a Final Context from it after resolving service conflicts. Finally, SPM provides Service Information and Service Status to Final Context Deliverer

V. EXPERIMENT

We first surveyed on how often family members conflict with each other in daily activities. We then evaluated the effectiveness of the conflict resolution method based on the ubiTV. The proposed conflict resolution method recommends several contents or services sorted by conflicting users' preferences when multiple users attempt to access a registered service. In addition, it selects one Media service when more than one Media service is simultaneously involved to the same resource. Finally, we conducted a survey on the usefulness of the conflict resolution method to family members.

A. Implementation

In order to implement the proposed Conflict Manager, we utilized ubi-UCAM 2.0. The ubi-UCAM 2.0 is a unified context-aware application model for ubiquitous computing environments supporting independence between sensors and applications [19]. Especially, the proposed Context Manager was implemented as a Context Manager of ubiService in the ubi-UCAM 2.0. The ubi-UCAM 2.0 also utilize unified context to represent and share users' contextual information among various applications. The unified context represents the contextual information as 5W1H (Who, What, When, Where, How and Why). We then applied the ubiService recommending contents and services to each Media service of ubiTV application such as television, movie, music, and Internet service, in ubiHome test-bed [26]. Figure 14 shows an example of service recommendation.



(a)TV screen (b) User's PDA

Fig. 14. Service recommendation

In this scenario, there are two users in ubiHome. They are in conflicts due to their preference on the television. User 2 wants

to enjoy News program, but user 1 is already enjoying Drama. Therefore, the Television service recommends available contents sorted by their preferences. Finally, the Television service recommends News, Drama, and Education programs on User 2's PDA and the large screen. After their discussion about television programs with recommended contents, they decide their proper program in this conflict situation.

Furthermore, to set up the service condition of each Media service, we conducted a survey on service preferences of users about their home environment. The survey was conducted for the home appliances frequently used in living room. Seventy persons, (40 parents / 30 children), were asked the following questions.

Question 1: What kind of services or appliances do you use frequently in your home?
Question 2: When do you usually use the services answered in the Question 1?

As a result, we found that parents usually spend their time on watching television around 9 P.M. Especially, they prefer to watch news to get social or weather information through the television. However, children usually spend their time by watching movie or using Internet. They also enjoy watching sitcom or comic programs through television. Based on their preference and time, we assigned conditional contexts for each user to services of ubiTV. Furthermore, for the experiment, the number of family members was three: two parents, and a son. This is the average family size in Korean family system aged from 30 to 40 [30].

In addition, in order to decide amount of conflict in home environment, we questioned 70 volunteers in ages from 10 to 60 who had experienced context-aware service supporting conflict resolution. They were asked to answer the following question.

Question 3: Who is the most related person when you are trying to use television service?

Table 9 shows the amount of conflict among family members based the survey.

Table 9. The amount of conflict among family members (Unit: %)

	Father	Mother	Children
Father	-	40	60
Mother	30	-	70
Children	33	20	47

As shown in Table 9, the conflicts appeared high in the viewpoint of parents when they were using the service with their children. In the case of children, they showed high conflict rate when they spend their time on using the service with siblings. This result implies that conflicts are occurred because the preferences of each family member are different in using services in home environment. Moreover, each member feels service conflicts differently. This is because the persons who are together are different with each other, when they spent their time on using the services in the living room.

B. Experimental Results and Analysis

In order to measure the usefulness of the resolution method of the proposed Context 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 worked. Table 10 shows hit ratio on the recommended genres of each users.

Table 10. Hit ratio on recommended genres (unit :%)

Users	Hits	News	Drama	Edu.	Ani.	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 10, the users show different characteristics on the recommendation. Father expressed higher selection on News program than that of other programs, but showed relatively low hit ratio 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 relatively higher hit ratio than others. However, they were unlikely to choose their preferred program soon after content recommendation. This is because Context 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 together.

In addition, in order to verify usefulness of recommendation of services, we configured properties of each Media service. In our experimental setting, all the Media services were on the same computer having a large screen and a sound device. Table 11 shows the properties given to each Media service.

Table 11. Property of Media services

Services	Resources
Television	{sound, display}
Music	(sound)
Internet	{display}
Movie	{sound, display}

As can be seen Table 11, each Media service has its own set of required resources, such as sound, display, or both, according to the resources it uses. Therefore, services which require the same resource cannot be executed simultaneously. Such service can start after stopping other services. For example, the television service uses sound, and display

resources and Internet service needs display resources. In this situation, those two services cannot be executed at the same time, because they share the display resource. In order to observe hit ratio on service recommendation, we monitored the Media services in ubiHome. Table 12 shows the hit ratio on recommendation of service conflicts among Media services given to different users.

Table 12. Hit ratio on recommended services (unit :%)

Users	Hits	TV	Music	Movie	Internet
Father	21	38	34	9	19
Mother	43	40	31	17	12
Son	55	30	16	25	29

As shown in Table 12, the users showed their characteristics on each service recommendation. Father shows relatively low hit ratio than that of other users. It means that he usually consents on a current service of other users. However, mother and son selected another service rather than enjoying a current service. In case of mother, the selections were made for the family. On the other hand, son's selections were made for himself. Nevertheless, we found that even though they changed the current service, they didn't choose their preferred program soon after the recommendation. They spent their time to talk about the current service and recommended services and then decided a suitable Media service.

Finally, in order to evaluate the satisfaction of the proposed conflict resolution method, we questioned following question to 30 volunteers in ages from 20 to 39 who had experienced ubiTV application supporting recommendation. We then compared the result with that of a resolution method selecting a user's context [25].

Question 1: What do you think of context-aware services that recommend users' conflicting services and contents when several members try to use them at the same time?

Table 13 shows the user satisfaction on the proposed resolution and the resolution method selecting one user.

Table 13. User satisfaction (%)

Age	User selection	Service recommendation
20~29	59	73
30~39	53	71

As shown in Table 13, the respondents showed higher satisfaction on the proposed resolution method than that of resolution method selecting a user with the highest preference. This is because selecting a user also causes another conflict, and remained users couldn't be satisfied to the resolution result. Therefore, users voted for the resolution method that recommends services or contents to resolve the conflicts. With additional information, they have a chance to talk about their interest and reach to solution.

VI. Conclusion

In this paper, we proposed the Context Manager to resolve service conflicts that arise when multiple users access a media services and when the media services are trying to share resources in the smart home environment. In order to resolve service conflicts among users, the proposed Context Manager recommended available contents ordered by preferences of conflicting users. The proposed Context Manager also detected conflicts among Media services by utilizing their service profile, and resolved them by selecting a Media service having the highest preference. Furthermore, the Context Manager resolved the conflicts among Media services occupied by different users by recommending conflicting Media services. Through the experiment on ubiTV application, we have shown the effectiveness of the proposed resolution method.

However, several problems are still remained. The proposed recommendation method needs more improvements to resolve conflicts, even though it gives users higher satisfaction than selection method. In some situation, the recommendation is unnecessary or has to give users more information. In addition, we have to observe the conflicts and resolution results because preferences of users are changing in time to time. Furthermore, we have to elaborate more on the usage of resources to accommodate various appliances and devices in home environments.

Acknowledgment

The authors would like to thank Seiie Jang for his valuable input and useful discussion about conflicts in context-aware applications.

References

[1] Mark Weiser. Computer of the 21st Century. *Scientific American*, 265(3): 94-104, September, 1991.

[2] Schilit, B., Adams, N. Want, R. Context-Aware Computing. *Proceeding of the 1st International Workshop on Mobile Computing System and Applications*, pp. 85-90, 1994.

[3] Anind K. Dey, "Understanding and Using Context. *Personal and Ubiquitous Computing*, Special issue on Situated Interaction and Ubiquitous Computing, 5(1), 2001.

[4] Neustaedter, C. and Greenberg, S. "The Design of a Context-Aware Home Media Space", *Proceedings of UBICOMP 2003 Fifth International Conference on Ubiquitous Computing*. 297-314, LNCS Vol 2864, Springer-Verlag.

[5] 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.

[6] Y.Oh, W.Woo, "A unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness," *Proc. Of Second Intern. Symp. On Ubiquitous Computing Systems (UCS2004)*, pp. 117-122, 2004.

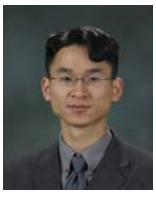
[7] Hughes, J., O'Brien, J., Rodden, T.: Understanding Technology in Domestic Environments: Lessons for Cooperative builds. *In Proceedings of the First International workshop on cooperative buildings (CoBuild'98)*. Darmstadt, Germany, Heidelberg, Germany: Springer-Verlag, pp. 246-261, 1998.

[8] W.Keith Edwards and Rebecca E. Grinter, "At Home with Ubiquitous Computing: Seven Challenges," LNCS 2201, pp. 256-272, 2001.

[9] Meyer, S. and Rakotonirainy, "A survey of Research on Context-Aware Home.", *Proc. Of the Australasian information serucrity workshop conference on ACSW frontiers 2003*, pp. 159-168, 2003.

[10] Christian Kray, Rainer Wasinger, and Gerd Kortuem, *Proceedings of the workshop on Multi-User and Ubiquitous User Interfaces (MU3I) at IUI 2004*, Funchal, Madeira, Portugal, ISSN 0944-7822, pp. 7-11, 2004.

[11] S.M. Easterbrook, E.E. Beck, J.S. goodlet, L. Plowman, M. Sharples and C.C Wood, "A survey of Empirical studies of Conflict", *CSCW: Cooperation of Conflicts?* London: Springer-verlag, pp. 1-68, 1993
[12] McCarthy, J.F. and T.D. Anagnost (1998): "Music FX: An arbiter of group preferences for computer supported collaborative workouts", in *Proceedings of CSCW '98*. Settle, WA: ACM Press.
[13] C.Shin and W.Woo, "Conflict Resolution among Users by Utilizing Context History", *Proceedings on ECHISE'05 held in Conjunction with PERVASIVE'05*, pp. 105-110, 2005.
[14] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton, "Building Agent-Based Intelligent Workspaces," In *ABA Conference Proceedings*, June 2002.
[15] 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.
[16] S. S. Yau, F. Karim, Y. Wang, B. Wang, and S.Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing," *IEEE Pervasive Computing, joint special issue with IEEE Personal Communications*, 1(3), , pp.33-40, July-September. (2002)
[17] Judd, G, Steenkiste, P., "Providing Contextual Information to Pervasive Computing Applications", *IEEE International Conference on Pervasive Computing (PERCOM)*, Dallas, March 23-25. (2003)
[18] John Canny, Danyel fisher, "Active-Based Computing," in *Proceeding of CHI, The Hague, The Netherlands*. (2000)
[19] S.Jang, and W.Woo, "Unified Context Describing User-Centric Situation: Who, Where, What, When, How and Why", *the 1st Korea/ Japan Joint workshop on Ubiquitous Computing and Network Systems*, 2005.
[20] Y.Oh, C.Shin S.Jang, and W.Woo, "ubi-UCAM 2.0: Unified Context-aware Application Model for ubiquitous computing environments", *the 1st Korea/ Japan Joint workshop on Ubiquitous Computing and Network Systems*, 2005.
[21] S.Jung and W.Woo, " UbiTrack: Infrared-based user Tracking System for indoor environment," *ICAT'04*, 1, paper 1, pp. 181-184, 2004
[22] Sj.Oh and W.Woo, "Manipulating multimedia contents with Tangible Media Control", LNCS(*ICEC*), vol.3166, pp.57-67, 2004
[23] Anand Ranganathan, Jalal Al-Muhtadi, Roy H. Campbell, Reasoning about Uncertain Contexts in Pervasive Computing Environments.. In *IEEE Pervasive Computing*, pp 62-70, Apr-June, 2004.
[24] Panu Korpipaa, Jani Mantyjarvi, Juha Kela, Haikki Keranen, and Esko-Juhani Malm, Managing Context Information in Movile Device, In *IEEE Pervasive Computing*, pp. 42-51, July-September, 2003.
[25] C.Shin, Y.Oh and W.Woo, "History-based Conflict Management for multi users and multi services", *Proc. of 1st Workshop on Context Modeling and Decision Support*, Paris, France, July 5, 2005, ISSN 1613-0073,Vol-144.
[26] S.Jang, C.Shin, Y.Oh, and W.Woo, A introduction of "ubiHome" Test-bed, *the 1st Korea/ Japan Joint workshop on Ubiquitous Computing and Network Systems*, 2005.
[27] Y.Suh and W.Woo, User Profile Management for Context-aware Applications in Smart Home Environments, *the 1st Korea/ Japan Joint workshop on Ubiquitous Computing and Network Systems*, 2005.
[28] D.Hong and W.Woo, A Toolkit for Context-aware Application in Wearable Computing, *the 1st Korea/ Japan Joint workshop on Ubiquitous Computing and Network Systems*, 2005.
[29] http://java.sun.com/products/personaljava/
[30] http://www.nsf.or.kr



Choonsung shin received the B.S degree in Computer Science from Soongsil University in 2004. Now he is a M.S. student in Department of Information and Communications, Gwangju Institute of Science and Technology (GIST) since 2004.
Research Interest: Context Awareness, Human Computer Interaction, Ubiquitous/ Wearable Computing.

Daeho Han received the B.S degree in Computer Engineering from Hanyang University in 2000 and M.S. degree in Computer Engineering from Hanyang University in 2002. During 2002-2005, he stayed in Samsung Advanced Institute of Technology (SAIT). Now he is Samsung Electronics Digital Solution Center, Seoul, Korea.
Research Interest: Operating System, Home Network, Ubiquitous Computing.

Woontack Woo received the B.S degree in Electrical Engineering from Kyungbook University in 1989 and M.S. degree in EE from Postech in 1991. He received Ph.D. in Electrical Engineering from university of Southern California. During 1999-2001, he stayed in ATR Media Integration & Communications (MIC) Laboratory of Japan as an Invited Researcher. Now he is Assistant Professor at Gwangju Institute of Science and Technology (GIST) U-VR Lab., Gwangju, Korea.
Research Interest: 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitous computing, etc.