

Conflict Resolution based on User Preference and Service Profile for Context-aware Media Services^{*}

Choonsung Shin and Woontack Woo

GIST U-VR Lab.
Gwangju 500-712, S.Korea
+82-(0)62-970-2226

{cshin, wwoo}@gist.ac.kr

ABSTRACT

This paper proposes a conflict resolution method for context-aware media services. The conflicts are caused by use of the service among multiple users and by share of the resources among various services. In order to resolve the conflicts among users, the proposed conflict resolution method sums the preferences of each user, and then recommends a customized service with the highest preference. In addition, it detects conflicts among services by exploiting their property, and determines their priority by utilizing preferences of users. Furthermore, to show the usefulness of the proposed conflict resolution method, we apply the proposed method to ubiTV service, context-aware media services, in ubiHome. Therefore, we expect the proposed method can play an essential role for context-aware media services to offer personalized services to multiple users by resolving service conflicts among applications as well as among multiple users.

Keywords: Context-awareness, context history, service conflicts, user preference, service profile

1. INTRODUCTION

Context-aware applications offer appropriate services to users by utilizing contextual information of environment including users. This information is obtained from various sensors or computing resources distributed in our daily life [1]. However, conflicts occur when multiple users share the applications or these applications share limited resources in the environment. Service conflicts occur when multiple users access services, and then the services have to choose one user to provide a customized response [11]. Finally, the applications could not make a suitable decision to start a service, and application users will not receive their personalized service. Also, resource conflicts among services happen if several services attempt to share the limited resources simultaneously. Therefore, applications

begin without possessing all the necessary resources and the users receive incomplete services.

Most research, aimed on resolving conflicts, has been done on smart home and intelligent office. Context Manager (CM) in ubi-UCAM selects a user and an application by assigning priority to users and applications [11]. ReBa (Reactive Behavioral System) supports conflict resolution among devices in office environment such as, between electric lamps, display devices, and telephones [2]. RCSM (Reconfigurable Context-Sensitive Middleware for Pervasive Computing), an object-based framework, makes sensors and application services independent, forms ad-hoc communication between them, and delivers the necessary context to the applications [2]. Context Toolkit collects, interprets, and delivers context between sensors and application services [3]. CIS (Contextual Information Service) manages contextual information, such as location and characteristics of users, devices, and status of network for providing contexts to applications [4][5].

However, most context management methods of the previous researches have various limitations when they are applied to multi-user environment with various applications. CM in ubi-UCAM gives satisfaction to only one user and one application [11]. In the case of ReBa, it is difficult to provide to each user with particular services because ReBa focuses on the service for grouped users by inferring main activities from the environment [2]. In RCSM, the context management method doesn't consider shared devices or services because contextual information services are provided only through individual device possessed by each user [3]. In the case of Context toolkit and CIS, application developers have to consider both conflict between services and between users because contexts are delivered when current context of environment matches an application-specified condition [4][5].

^{*} This work was supported by DSC of SEC in Korea.

In this paper, we propose Conflict Manager to resolve conflict caused by use of services among multiple users and limited resources among multiple services. It consists of three parts: conflict resolution among users, conflict resolution among services and service profile management. In case of conflict resolution among users, the proposed Conflict Manager detects and resolves conflicts caused by utilizing users' preference. It also detects conflicts among applications by utilizing service information and resolves the conflicts by exploiting users' preference.

Therefore, the proposed conflict resolution method has following benefits. Firstly, it recommends the most preferred content of the registered service to users within a service area. It also selects the most preferred service among various conflicting services within a service area. Furthermore, the proposed resolution method resolves conflicts without additional storage to keep tracks of user history. Therefore, the proposed resolution method gives users higher satisfaction than the resolution method selecting user's service.

This paper is organized as follows. In Chapter 2, we introduce unified context-aware application model for ubiquitous computing environments. Chapter 3 describes conflict resolution method to resolve conflicts among multiple services and among multiple users. In chapter 4, we explain how to manage service profile. Experimental setup and results are shown in Chapter 5. Finally, we discuss conclusions in Chapter 6.

2. Unified Context-aware Application Model

Before dealing with conflict problems, we have to consider how to represent contexts and how to be aware of contexts. Thus, we introduce context representation (ubi-UCAM), context-aware application model, and context management.

2.1 Context Representation

In order to represent context information in user-centric service, we utilize a unified context [6]. It represents context information with 5W1H(Who, What, Where, When, How, Why) which is used to express information about user context comprehensively while including some part of context information about surrounding environment in order to provide the personalized service. With each field having sub-fields, the unified context also express as context about a user in detail. The unified context expressed with 5W1H ensures independence between sensors and services, and it takes advantages being re-used in other services. Especially, the unified-context format ensures the simultaneous connectivity between sensors and a lot of services, reducing additional management which needs to change the same context to other form according to individual services.

2.2 Unified Context-aware Application Model

To provide users with the personalized service, we exploit

ubi-UCAM (Unified Context-aware Application Model for ubiquitous computing environment) which is a context-aware application model [6]. It utilizes various interactions that use context in a ubiquitous computing environment where many different kinds of sensors and services are distributed to environments. Ubi-UCAM, illustrated in the Figure 1, is composed of ubiSensor and ubiService.

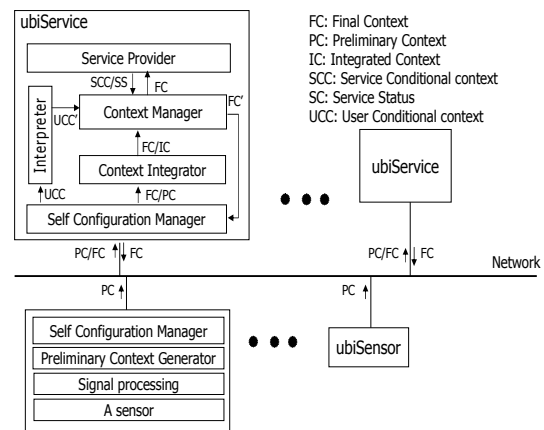


Figure 1. ubi-UCAM

As can be seen in the Figure 1, a ubiSensor is composed of a Sensor module to perceive a user and changes of user environment, and Preliminary Context Decision module to generate preliminary context with the perceived information. A ubiSensor generates and delivers a preliminary context which has all or part of the 5W1H. A ubiService is composed of Context Integrator, Context Manager, Interpreter and Service Provider. Context Integrator collects preliminary context from ubiSensors within the same working area in a given time interval and classifies 5W1H to each sub-element. It also analyzes its characteristics by each item of the collected 5W1H by applying a decision making technique to generate integrated context. Context Manager searches conditional context corresponding to the integrated context from a hashtable, which manages specific service actions and conditions, and generates final context which is to be used in applications. Finally, Service Provider executes proper action with parameters based on the final context.

2.3 Context Management

In the ubi-UCAM, the Context Manager mainly focuses on the ability to resolve conflicts which must be solved in context-aware media services with multiple users and multiple services. In addition, Context Manager uses short-term history of users and services to solve conflicts dynamically. To deal with conflict with dynamic approach, Context Manager carries out context-preprocessing, conflict resolution with history management, and final context generation.

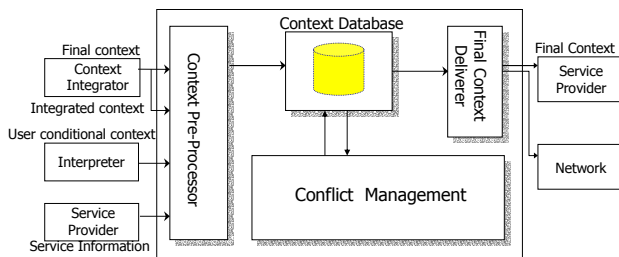


Figure 2. Context Manager

Context Preprocessor carries out pre-treatment about the integrated contexts in order to use Context Manager. To do this processing, it deals with integrated context, conditional context, and final context.

Context Database keeps necessary contexts and related information to support generation of final context. It contains several kinds of context, such as conditional context, final and user context within a service area, and conflict history of users and a registered service

Conflict Management involves not only conflict resolution among users and services, but also weight learning to dynamically determine priority of users and services.

Final Context Deliverer provides Service Provider and Self Configuration Manager with the final context which has service action and information about a user who use it.

3. Conflict Management

Conflicts of context-aware media services occur not only due to multiple users who access the services at the same time. It also happens by services trying to use shared resource in their surrounding. Figure 3 shows the overall architecture of Conflict Manager.

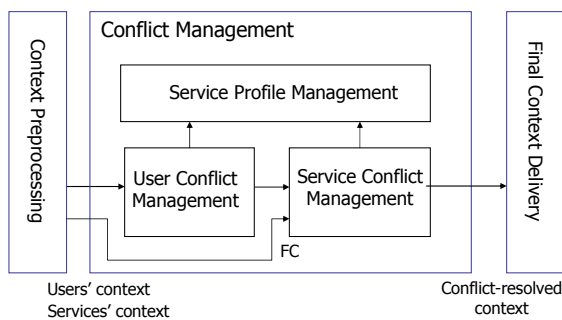


Figure 3. Conflict Manager

As show in Figure 3, Conflict Manager consists of User Conflict Manager, Service Conflict Manager and Service Profile Manager. Conflict Manager exploits each preference of users, and selects one content with the highest preference among them. In addition, to deal with conflict among services, based on the properties of services and relationship between them, Context Manager detects and resolves conflict. Therefore, Conflict Management involves not only conflict resolution among users and services, but also manages service profile.

3.1 Conflict Resolution among Users

Conflict Manager manipulates user contexts in two steps: building a user conflict list and selecting a proper user from it. In the former, Conflict Manager makes a conflict list of context on users who are expected to cause conflict among users, including those who are currently using the service. In this process, users who leave out of the service area are excluded in the list because we assume they don't want to use the service any more. In the next stage, Conflict Manager selects a content having the highest preference among users. Figure 4 shows preference selection algorithm in Conflict Manager.

```

Int Selected_Content.
Begin i = 0 to the number of users with in a service area
  Begin j = 0 to the number of contents in a Service
    Preference[j] =
      Preference[j] + User[i].Who.Preference[j]
  End
End
Selected_Content =
  Max (Preference[0], Preference[1], ..., Preference[N])

```

Figure 4. Preference selection algorithm

In this algorithm, Conflict Manager adds each preference of users. It then selects a content having the highest preference. Figure 5 shows an example of the service conflict among users.

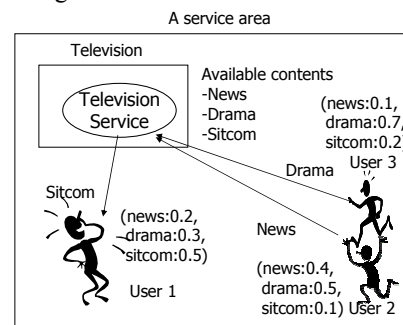


Figure 5. Conflict resolution among users

As shown in Figure 5, there is a television service providing user 1 with sitcom program in a service area. Simultaneously two users, user 2 and user 3, are trying to use the service. Therefore, a service conflict arises due to use of television services by the three users. In this conflict situation, Conflict Manager builds a conflict list consisting of contexts of user 1, user 2 and user 3. Based on the conflict list, it then sums the preferences of each user. In this scenario, we user 1 has preference on contents of television: 0.2, 0.3, and 0.5 to News, Drama, and Sitcom, respectively. User 2 has preference on contents of television: 0.4, 0.5, and 0.1 to News, Drama, and Sitcom, respectively. User 3 has preference on contents of television: 0.1, 0.7, and 0.2 to News, Drama, and Sitcom, respectively. User Conflict Manager obtains the preference

of 0.7, 1.5 and 0.8 on the contents of the service. Consequently, Conflict Manager recommends a Drama program having the highest preference as a content for television service.

3.2 Conflict Resolution among ubiServices

In addition, Conflict Manager resolves conflicts caused by multiple services which are trying to share limited resource in a service area. Figure 6 shows conflict resolution among ubiServices. At first, Conflict Manager resolves conflicts caused by other services within a service area. It creates context which contains information about the service and its stop command, if resources involved in other services are the same as those of the service itself. As a result, the ubiService responds to changes of other services which cause conflict, using final contexts which come from other services. Conflict Manager also prevents the registered service causing conflict with other services. To detect possible conflicts, it checks if there are any services exploiting the same resource before delivering the context. Finally, Conflict Manager sends the context to Final Context Deliverer when there aren't any services related to the same resource. Otherwise, Conflict Manager stores the context.

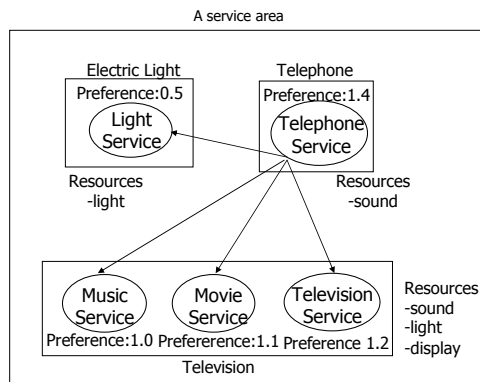


Figure 6. Conflict Resolution among ubiServices

As shown in Figure 6, there are three devices: electric light, television and telephone. Electric light provides light service, telephone offers telephone service and television shows movie, music, and television services. These services also utilize specific resources of each device. In this situation, the telephone service causes a conflict with television service due to the sound resource. Therefore, Conflict Manager detects the conflict in inward conflict resolution. It then builds a final context containing a stop command for the registered service. Afterward, Conflict Manager compares the preferences of the registered services with the preferences of telephone service. According to preference, it selects the context of the telephone service.

4. Service Profile Management

In order to resolve conflicts among users and ubiServices, Conflict Manager manages users' preference and service information. In order to manage the registered service, Conflict Manager obtains service information from Service Provider. The information includes various properties of the service, such as location, property, identity, etc. It also keeps a track of a user who current uses the registered service. Figure 7 shows the service information managed in Conflict Manager.

Table 1. Service Information

Category	Information
A user (5W1H)	A user who currently uses the registered service.
The registered Service (5W1H)	Types of the service Current status of the service Service name A service area Time when the service started

Firstly, Conflict Manager requests properties of the registered service from Service Provider. The properties describes characteristics of the registered service, such as display, sound, light service, etc. the properties are used to detect conflicts among ubiServices. For example, television service has three properties, sound, display, and light. The current status of a service shows a current state of the service. These includes "on", "off", "recommend", "pause", etc. These are also used to resolving conflicts among ubiServices. The service name enables Conflict Manager to identify each ubiService within a service area. The service area of the registered service makes a boundary of service conflicts. The time maintains time information when the user started to use the registered service. The user context describes a user who current use the service

5. Implementation and Experiments

In order to apply the proposed resolution method to various applications, we implemented Conflict Manager with J2SDK 1.4. As shown in the Figure 7, we applied the Conflict Manager to ubiHome test-bed [9]. In the ubiHome, we tested the Conflict Manager with ubiTV service, context-based media services in smart home environments. It provides users with various services such as, a television service, an Internet service, a music service, a movie service, etc. the ubiTV offers customized response to users. In addition to these context-aware services, we also exploited various sensors, such as couch sensors to detect user's behaviors, ubiTrack to track user's location [10], and remote controllers to control the services.

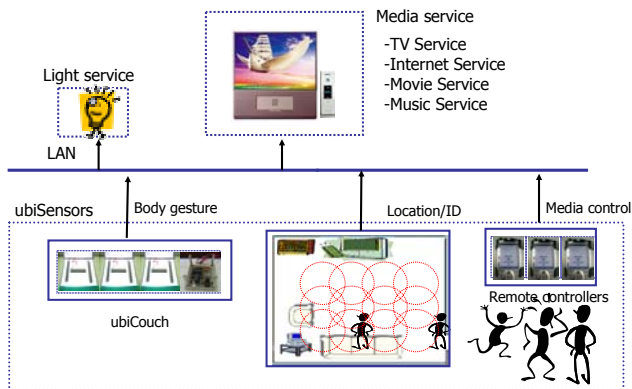


Figure 7. Context-based Media services

In order to measure the satisfaction of users, we experimented on service conflicts in two ways: one is the resolution method recommending a customized service and the other is the resolution method selecting one user. The proposed resolution method selects a content of the service with the highest preference. The resolution method assigns priority to conflict contexts and then chooses one user having the highest priority when conflict occurs [11]. To test two methods, we employed a television service which is widely used in home environments. While using the television service, family members cause conflicts due to their preferences and its broadcasts. In our experiment, the television service recommends a program having the highest preference based on each resolution strategy.

In addition, we configured properties of ubiServices to deal with conflict among services. In the experiment, all the ubiServices were in the same area. Especially, television, movie, Internet and music services were operated on the same computer. We then monitored the ubiServices in ubiHome. Table 2 shows the properties given to each ubiService.

Table 2. Property of ubiServices

Services	Sound	Display	Light	Location
Television	O	O	O	Living room
Internet		O		Living room
Music	O			Living room
Movie	O	O	O	Living room

In case of television service, most conflicts are related to Internet service. The rest of the conflicts are associated with movie and music service. Movie service, which shares sound, light, and display resource, is related to all the services. In particular, conflicts with Movie service are mostly due to TV service which is accessed by user. Besides, Internet and music service were related to television and Movie service using sound and display resources. Therefore, conflicts between these services depend on users and their preference of using services in home environments.

Furthermore, to evaluate the effectiveness of the proposed conflict resolution method, we asked twenty persons, who experienced the proposed resolution method, to answer following question.

Question: What do you think of context-aware services that recommend a proper content when several members try to use them at the same time?

Figure 8 shows the user satisfaction on the proposed resolution and the resolution method selecting one user.

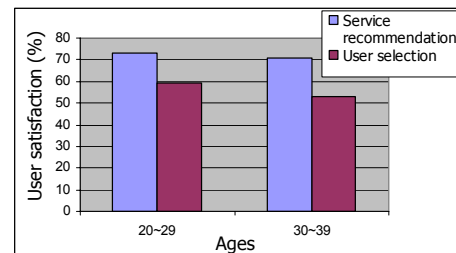


Figure 8. User satisfaction

As shown in Figure 8, the respondents showed higher satisfaction on the proposed resolution method than that of resolution method selecting one user. The people aged in 20s showed 73 percent on the proposed resolution method. However, they expressed 59 percent on the selecting method. In case of people aged in 30s, they show 71 percent on the proposed resolution method. On the other hand, they express 53 percent on the selecting method. Therefore, we found that most users preferred to the proposed resolution method recommending favorite contents rather than selecting one user having the highest priority. This is because the users want to use their services continuously when they hold their services.

6. Conclusion

In this paper, we proposed the conflict management method to resolve conflicts that arises when multiple users access various context-aware applications in ubiquitous computing environments. In order to resolve conflicts among users, the proposed Conflict Manager maintains contextual information of users and service profile, such as types of the registered service, location, time, and then selected one service most user have preference among various each service. In addition, Conflict Manager detects conflicts among services based on the properties of each service. It resolves the conflict by selecting one ubiService with the highest preference among conflicting ubiServices. In the experiment, we found that users showed higher satisfaction on the proposed conflict resolution method than conflict resolution method selecting one service. For future works, we plan to increase the number of users who

are involved in conflicts, and observe user's behaviors over longer periods.

Acknowledgement

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

REFERENCES

- [1] Anind K. Dey, "Understanding and Using Context. Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing, 5(1), 2001Lampport, L. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley, Reading, MA, 1986.
- [2] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton, "Building Agent-Based Intelligent Workspaces," *In ABA Conference Proceedings*, June 2002.
- [3] 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), July-September 2002, pp.33-40.
- [4] 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.
- [5] Judd, G, Steenkiste, P., "Providing Contextual Information to Pervasive Computing Applications", *IEEE International Conference on Pervasive Computing (PERCOM)*, Dallas, March 23-25, 2003.
- [6] S.Jang, and W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model", *Lecture Note Artificial Intelligence*, Vol, 2680, pp.178-189, 2003.
- [7] Garcia-Molina, H.(1982). Elections in Distributed Computer Systems. *IEEE Transactions on Computers*, Vol, C-31, No. 1, pp. 48-59.
- [8] Jiawei Han, Micheline Kamber, (2001). *Data Mining: Concepts and Techinques*, Morgan Kaufmanm.
- [9] Y.Oh, W.Woo, "A unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness," *UCS2004*.
- [10] S.Jung, W.Woo, " UbiTrack: Infrared-based user Tracking System for indoor environment," *ICAT'04*, paper 1, pp. 181-184, 2004.
- [11] C.Shin and W.Woo, "Conflict Resolution among Users by Utilizing Context History", *Proceedings on ECHISE'05 held in Conjunction with PERVASIVE'05*, pp. 97-101, 2005