The Design and Implementation of a Context-Aware Group Communication System

Chichang Jou
Information Management Department
Tamkang University
Tamsui Taipei, Taiwan

Wei-Jiun Wang
Information Management Department
Tamkang University
Tamsui Taipei, Taiwan

Abstract—*Past researches on context-aware computing were mostly focused on supporting individuals, and lacked supports for group members. The few researches of context-aware group communication only allowed group members in the same locality to interact. We design and implement a context-aware group communication system, called CAG, which can track the mobile users' current locations, provide basic group communication services, and help mobile users interact with their group members. The CAG system could provide services for group members in the same or different localities. It is applied in a scenario regarding groups in an exhibition, and its performance is evaluated.*

Keywords: Context-aware computing, Mobile computing, Group communication service, Pervasive computing

1 Introduction

With the wide spread deployment of wireless networks and universal installation of wireless connectivity firmware in handheld devices, pervasive computing [9][10] has become a very important mobile computing paradigm. The goal of pervasive computing is to obtain and respond to related data collected from devices anywhere at any time. The challenge to achieve this goal is how to recognize and adapt to the changes in the surroundings [11]. This requirement spawned the emerging research realm of context-aware computing, which was defined by Schilit and Theimer [12] as software that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time."

Past researches on context-aware computing were mostly focused on supporting individuals. For example, the Active Badge Location system [13] and the RADAR system [1] have been concentrated on tracking and providing services to individual mobile users. As far as we know, the only studies of group member support in context-aware computing were the AGAPE (Allocation and Group Aware Pervasive Environment) middleware [2] and the GUIDE project [5]. They did utilize the context information to support interactions among group members. However, they only allowed group members in the same locality to interact. We design and implement a context-aware group communication system (abbreviated as CAG). The CAG system utilizes context information of mobile users, such as their identities, their current locations, their time-stamped activities, to support communication and interaction for group members. It could provide services for group members in the same or different localities.

The remainder of this paper is organized as follows: Section 2 surveys the definitions of context in the literature, and then reviews current group communication support in context-aware computing. In Section 3, we describe the overall operating environment of the CAG system, analyze its required functionalities, and then design the architecture and components of CAG. Section 4 illustrates how we implement CAG and its group services. In Section 5, we apply the CAG system in a scenario regarding groups in an exhibition, and evaluate its performance. Finally, in Section 6, we present conclusions, limitations and future works of this research.

2 Related work

In the early stage of context-aware computing, most researchers defined context according to their research needs. Schilit and Theimer [12] referred to context as location, identities of nearby people and objects, and changes to those objects. Brown *et al.* [4] defined context as location, identities of the people around the user, the time of day, season, temperature, etc. Ryan *et al.* [8] defined context as the user's location, environment, identity and time. Because most definitions are too *ad hoc* to be applied to other applications, Dey and Abowd [7] integrated them and proposed the following widely employed definition of context: "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

Dey and Abowd [7] categorized contexts into the following four primary context types: location, identity, time, and activity. The data of the primary context can then be used as indices to find detailed information, called secondary context, for the same entity, and to find the primary context of other entities. For instance, given a person's identity, many related detailed information of this person, such as his/her phone numbers, address, email address, relationship to other people in the environment, etc., could be acquired. With an entity's location, we can determine what other people or objects are near the entity and what activity is occurring near the entity.

As far as we know, the only two researches about group communication support in context-aware computing are the AGAPE middleware [2] and the GUIDE project [5]. Bottazi *et al.* proposed the AGAPE framework, which is location-aware and provides a set of group membership support facilities for collaborative applications. The notion of locality, defined as the set of entities in a location, is used in AGAPE to support group management. The entities in one locality may belong to the same or to different groups. A group is partitioned into disjoint subsets in which co-located group members in a locality are assembled. Based on the AGAPE framework, Bottazi *et al.* [3] further established a context-aware communication model, which manages communication through the characteristics of the communicating parties, such as their location and their profiles. Collaboration in this model is based on the metaphor of group. However, the interactions among group members are restricted to the entities within the same locality.

Cheverst *et al.* [5] developed web-based GUIDE system to provide electronic context-aware tourist guide through a handheld device. The system comprises a number of interconnected cells. Each cell is associated with a Linux-based cell-server, which acts as a proxy with cached local storage of the central GUIDE web server. The server is responsible for broadcasting information to the handheld devices when they enter some cell-server's coverage area. Cheverst *et al.* [6] extended the GUIDE system to support cooperation of city visitors so that they could make a remote request for a list of GUIDE users currently in a particular cell. The web server responds to the query by returning the identity of those visitors currently recorded as being present at the specified location. Then the visitors could read the opinions of others who had come to the same location, or discuss with those who are currently in the same cell. In the system the following visitors' contexts are used to support the interaction of city visitors: locations, identities, and profiles. The interaction, however, is confined in the same cell.

3 Architecture and System Analysis

To enable group members to interact beyond the scope of a single area, we design and implement the CAG system, which consists of one CAG maintenance server and several CAGs. The CAG system provides group communication services for mobile users by utilizing their related context to interact with their group members in both the same and different area via the collaboration among CAGs. This section illustrates the overall operating environment of CAGs and the components of CAGs.

3.1 Architecture of the CAG System

We adopt the cell-based wireless network infrastructure used in the GUIDE system. The covered computing environment is divided into several areas, herein called *domains*, in which one CAG is in charge of. Each CAG is connected with the backbone network through fixed network, as displayed in Figure 1.

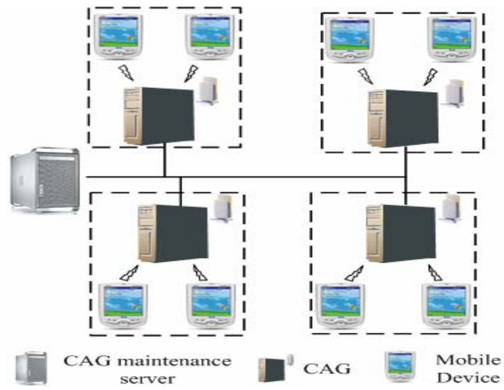

Fig. 1.Architecture of the CAG System

To provide the DHCP service to the mobile devices in its covered area, each CAG makes use of a Wi-Fi Access Point (AP). The CAG maintenance server maintains the IP addresses of the CAGs and their responsible domains. When a mobile user enters a CAG's domain, his/her handheld device senses the beacon sent by the Wi-Fi AP of the CAG and achieves physical layer connection with it. Then, the CAG automatically assigns an IP address to the device via the DHCP protocol so that the user could register with the CAG. At the time of registering, the CAG acquires and stores contexts related to the user. After completing the registration, the user can exploit the group communication services provided by the CAG system. This infrastructure provides the following two advantages: (1) Each CAG can identify all mobile users within its responsible domain with the Wi-Fi AP. (2) Each CAG can provide mobile users with group communication services over TCP/IP.

3.2 System Analysis of the CAG System

We identify several example group communication services and their required contexts in this subsection. We first identify the following group communication services (group services, for short) to support the interaction among mobile users:

1) *Registration Service* enables mobile users to register with a CAG.

2) *Search Service* enables mobile users to search their group members.
3) *Message Service* enables mobile users to send/receive messages to/from their group members.
4) *File Transfer Service* enables mobile users to transfer files to their group members.
5) *File Downloading Service* enables mobile users to download the temporarily stored files from CAG.

Secondly, we define the required contexts, based on the definition proposed by Dey and Abowd [7], for the above group services. The adoption of this definition is because it is widely accepted and it covers so far the maximal number of contexts for an entity. For example, we would like to make use of the activity context so that when a user could not accept real-time file transfer, he could download the file later on. In the CAG system, context is defined as "any piece of information which is needed when a CAG executes a group communication service". The contexts are also further classified to primary and secondary contexts listed in Tables I and II.

TABLE I
PRIMARY CONTEXT USED BY THE CAG SYSTEM

Category	Primary Context	Description
Location	Locality	The scope of searching or of sending messages to group members
Location	Position	A mobile user's current position
Identity	SessionID	The identification of the connection channel between a CAG and a handheld device
Identity	UserName	The identification of a mobile user in the responsible domain of a CAG
Time	RecvTime	The timestamp of receiving a file
Time	CurrentTime	The timestamp of downloading a file
Activity	UserStatus	The activity status of a mobile user

TABLE II
SECONDARY CONTEXT USED BY THE CAG SYSTEM

Secondary context	Description
GroupID	The group identification of a mobile user.
Interest	The interests of a mobile user.
IPAddr	The IP Address used by the handheld device of a mobile user.
FileName	The name of a file

3.3 The Components of a CAG

In Figure 2, we display the components of a CAG. The detailed interactions among these components are described in Section 4. We briefly introduce these components as follows:

1) *Data Storage* (DS) is the storage space of a CAG for storing mobile users' related contexts, consisting of the following three subcomponents: 1. Database: each CAG maintains the context database for all mobile users in its domain. 2. Repository: this is a file directory for storing temporarily files, which could be either a mobile user's un-received files or files from mobile devices and other CAGs. 3. Cache: it records the current locations, represented by the IP address of the responsible CAG for the domain, of mobile users' group members in a hash table.
2) *Registration Component* (RC) takes charge of the Registration Service.
3) *Searching Component* (SC) performs the Search Service, which has constraints on the scope of packet delivery listed in Table III.
4) *Messaging Component* (MC) performs the Message Service, which also has constraints on the following scope of packet delivery listed in Table III.
5) *File Transfer Component* (FTC) performs the File Transfer Service and the File Downloading Service.
6) *Application Interface* (AI) acts as the communication channel between the handheld device and a CAG after the device achieves physical layer connecting with the CAG.
7) *Collaboration Component* (CC) acts as an interface of the CAG with other CAGs. A CAG could obtain a "CAG list" from the CAG maintenance server, and then communicate with each CAG on the list through multi-threads in parallel.

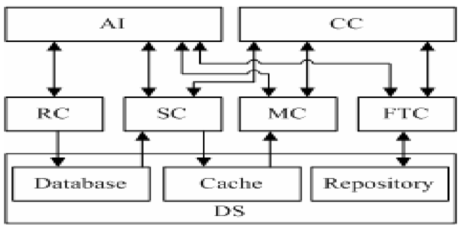

Fig. 2. The Components of a CAG

TABLE III
THE DESCRIPTION OF SEARCH AND MESSAGING SCOPE

Delivery scope	Description
ALL	All group members
ANY	No restriction on group
VICINITY	Location restricted to the responsible domain of a CAG
FLOOR	Location restricted to the same floor inside the domain of a CAG
BUILDING	Location restricted to the same building inside the domain of a CAG

4 System Implementation

We use HP iPAQ hx2410 Pocket PCs as the handheld devices and install CAG client application software in PersonalJava on them. Their operating system is Windows Mobile Second Edition, and they are equipped with a built-in Wi-Fi Ethernet module.

The platforms employed by CAGs and the CAG maintenance server are the same. Their operating system is Fedora Core 2 Linux, their development programming language is J2SE, and their database is MySQL v4.0.21. Since the implementation of the CAG maintenance server is quite straightforward, we skip its implementation details in this article. To support the pervasive computing, in each CAG we set up the DHCP and NAT services by ISC DHCP v3.0 and iptables v1.2.9 respectively. A Wi-Fi AP is also installed to construct the wireless network for a handheld device to obtain an IP address to connect to the CAG system. The following subsections describe the interaction of the components of a CAG for each group service.

4.1 Registration Service Implementation

The detailed operations of the Registration Service are illustrated in Figure 3, which are explained as follows:

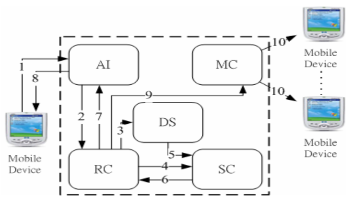

Fig. 3. Detailed Operations of the Registration Service

1) Steps 1 and 2: A handheld device sends the UserName primary context and the following secondary context: GroupID, Interest, and IPAddr to start the registration with a CAG.
2) Step 3: The Registration Component generates a unique SessionID for this connection and stores it to the Database, together with the context related to the user.
3) Steps 4 to 8: The Searching Component searches the user's group members from the Database and passes the search result to the Registration Component. The Registration Component marshals the result, and then sends them with the SessionID back to the device.
4) Steps 9 and 10: The Messaging Component sends a "group member entrance" message to all the user's group members in the domain so that all his/her group members are aware that the user is in the same domain.

4.2 Search Service Implementation

The detailed operations of the Search Service are illustrated in Figure 4, which are explained as follows:

1) Steps 1 and 2: A handheld device sends the Locality and UserName primary contexts to the CAG for searching. It also could send the Interest secondary context for the search.
2) Step 3: The Searching Component asks the Collaboration Component to inquire other CAGs.
3) Steps 4 and 5: The Collaboration Component obtains a "CAG list" from the CAG maintenance server. It then sends a broadcast packet, including the UserName primary context and/or the GroupID or Interest secondary context, to the CAGs on the list for searching the user's group members.
4) Steps 6 to 9: After each remote CAG received the broadcast packet, its Searching Component utilizes the related contexts to search mobile users from its Database. Then, its Collaboration Component returns the search result to the original CAG.
5) Steps 10 to 13: The Collaboration Component of the original CAG gathers the search results from all remote CAGs, and then passes them to the Searching Component. Then, the Searching Component marshals the results and stores them to the Cache. Finally, the original CAG returns the result to the user.

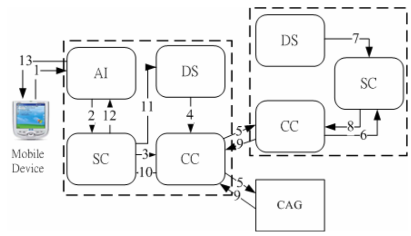

Fig. 4. Detailed Operations of the Search Service

4.3 Messaging Service Implementation

The detailed operations of the Messaging Service are illustrated in Figure 5, which are explained as follows:

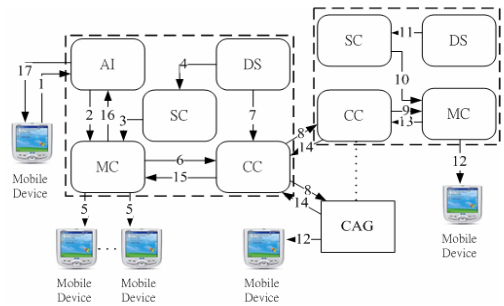

Fig. 5. Detailed Operations of the Messaging Service

1) Steps 1 and 2: A handheld device sends the Locality (i.e. the scope limitation in Table III) and receivers' UserName list, as well as the message to the CAG.
2) Steps 3 to 5: The Messaging Component determines whether the scope is VICINITY. If yes, the Messaging Component sends the message to the user's group members in current domain via the Searching Component out of the Database. Otherwise, it first sends the message to the group members in the receivers' list and in current domain. For group members in other domains, it performs Steps 6 to 17.
3) Steps 6 to 8: The Messaging Component passes GroupID, the message receiver list, the user's name and the message to the Collaboration Component. The Collaboration Component obtains each message

receiver's current location and sends a broadcast packet, containing GroupID, the sender's and the receiver's names and the message, to each remote CAG.

4) Steps 9 to 14: After each remote CAG has received the broadcast packet, its Messaging Component tries to find the receiver via its Searching Component. If the receiver is in its responsible domain, its Messaging Component sends the message to the receiver, and then returns a success packet to the original CAG. Otherwise, a failed packet is returned to the original CAG.
5) Steps 15 to 17: The Messaging Component of the original CAG removes the receiver from the list after the message is accepted successfully, and then marshals the messaging results. Finally, it returns the result to the handheld device.
6) Steps 6 to 17: If after the first round, some receivers still remain in the list, then the Messaging Component updates the Cache via the Searching Component and carries out the second messaging delivery.

4.4 File Transfer Service Implementation

The detailed operations of the File Transfer Service are illustrated in Figure 6, which are explained as follows:

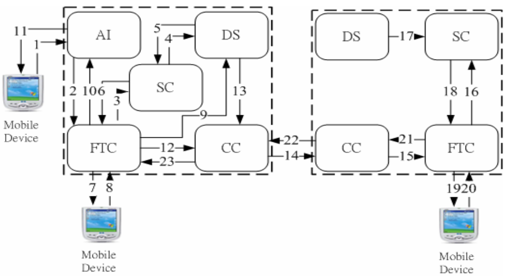

Fig. 6. Detailed Operations of the File Transfer Service

1) Steps 1 and 2: A handheld device sends the SessionID, the file receivers' UserName list and the FileName secondary context to the CAG. The File Transfer Component creates in the Repository an exclusive directory with the name of the user's SessionID, and then receives the file from the device.
2) Steps 3 to 6: After receiving the file successfully, the File Transfer Component obtains the receiver's SessionID via the Searching Component, and then creates an exclusive directory for the receiver.
3) Steps 7 and 8: the File Transfer Component copies the file from the sender's directory to the receiver's. Then, it informs the receiver and obtains his/her activity status.
4) Steps 9 to 11: If the status shows that the receiver could receive the file, the File Transfer Component transfers the file immediately. Otherwise, it stores the receiver's SessionID, the file name and the received time to the Databases so that the receiver can download it later.
5) Steps 12 to 14: If the receiver is not in the same domain, the File Transfer Component passes the related context to the Collaboration Component. After obtaining the receiver's current location and GroupID from the Cache and the Database respectively, the Collaboration Component sends a broadcast packet, including the related information, to the remote CAG.
6) Steps 15 to 23: The Collaboration Component of the remote CAG passes the packet to its File Transfer Component. Its File Transfer Component repeats the actions from step 3 to 11 after it received the file from the original CAG.

4.5 File Downloading Service Implementation

The detailed operations of the File Downloading Service are illustrated in Figure 7, which are explained as follows:

1) Step 1 and 2: A handheld device sends the SessionID primary context and the FileName secondary context to the CAG.

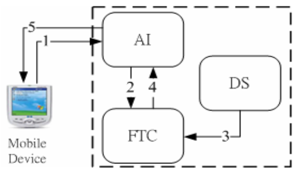

Fig. 7. Detailed Operations of the File Downloading Service

2) Step 3 to 5: The File Transfer Component compares the current time with the received time to determine, whether the elapsed time of the expected file has exceeded the maximum period of preservation (e.g. 30 minutes) specified by the CAG. If not, the File Transfer Component takes the user's temporarily stored file out of the Repository, and then sends it to the device. If yes, the CAG has already deleted the file beforehand. It then informs the user the bad news.

5 Application of the CAG system

We describe an application scenario of the CAG system in this section. The CAG maintenance server and four CAGs, named C71, C72, C91 and C92 respectively, are setup first. Suppose a group of sixteen people are visiting an Electronics Exhibition and they are dispersed in disjoint regions managed by the four CAGs, as listed as Table IV.

Figure 8(a) indicates that, user C7101 has registered with C71 and obtained the list of group members in the same domain, displayed in the "Vicinity Member" column. Figure 8(b) shows C7101 utilized the Interest secondary context to find out group members interested in "Program" and within the same building. Figure 8(c) illustrates that C7101 sent a message to all group members in the same domain. Figure 8(d) shows that C7101 sent the file named DSC0135.JPG to C9201 successfully. Figure 8(e) indicates that C7101 can not receive the file ant.jpg sent by C7201 immediately. The filename was listed in the "Unreceive File" column. Figure 8(f) shows that, C7101 exceeded the maximum preservation period to download the selected file.

TABLE IV
THE LIST OF GROUP MEMBERS' LOCATIONS

CAG name	Responsible domain	UserName
C71	1st exhibition area, 7F, Building B	C7101, C7102, C7103, C7104
C72	2nd exhibition area, 7F, Building B	C7201, C7202, C7203, C7204
C91	1st exhibition area, 9F, Building B	C9101, C9102, C9103, C9104
C92	2nd exhibition area, 9F, Building B	C9201, C9202, C9203, C9204

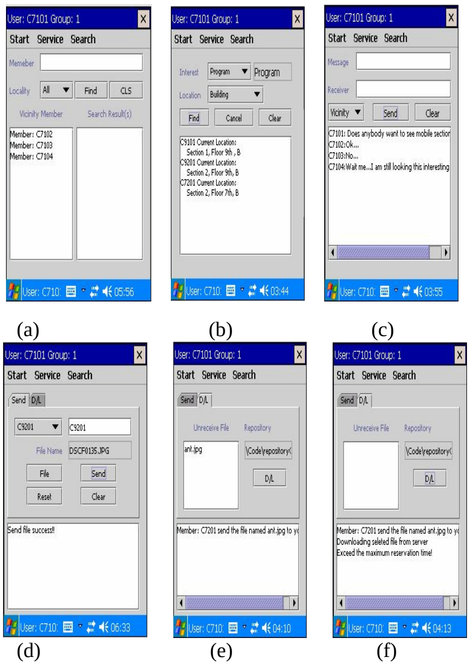

Fig. 8. The Snapshots of (a) Registration Service (b) Search Service, (c) Messaging Service, (d) the File Sender, (e) the File Receiver (Busy Status), and (f) Timeout to Download File

We list in Table V the average elapsed time for executing the group services 20 times. The time is from the CAG receiving the specified service to the CAG returning the results to the user, including the collaboration time among CAGs. Searching and sending a message to all group members take more time than other group services, because the local CAG has to interact with all other CAGs. Searching group members by the secondary context is usually with the constraints on the delivery scope, such as within the same region, floor or building. It needs less number of interacting CAGs. Therefore, it takes less time than the first two group services. The File Transfer Service of a 100K byte file takes the least elapsed time, since the local CAG could obtain the group member's locations from the Cache. It could respond immediately without inquiring other CAGs. Thus, the majority of elapsed time was spent in transferring and receiving a file. However, the elapsed time of the File Transfer Service depends on the size of the file.

TABLE V
THE ELAPSED TIME ON EXECUTING GROUP SERVICES

Group communication service	Elapsed time (ms)
Sending a message to all group members	750
Searching all group members	750
Searching group with secondary context	687
Transfer of a file with size 100 KB	546

We measure the elapsed time on the related components of a CAG in searching all group members and in sending a message to all group members. Tables VI and VII show that, the majority of the elapsed time are spent in the collaboration among CAGs. Although we implemented the collaboration process through multi-threads, network I/O events intrinsically take time.

TABLE VI
THE ELAPSED TIME OF COMPONENTS IN SEARCHING ALL GROUP MEMBERS

Component	Elapsed time (ms)	Description
Searching component	718	The time of searching from receiving the request to returning the results
The CAG maintenance server	125	The time of getting a CAG list
Collaboration component	578	The time of interacting with other CAGs
Data Storage	0.5	The time of storing the search result to the Cache

TABLE VII
THE ELAPSED TIME OF COMPONENTS IN SENDING A MESSAGE TO ALL GROUP MEMBERS

Component	Elapsed time (ms)	Description
Messaging component	781	The total time of sending a message to returning a success message
Searching component	47	The time of searching vicinity group members from their databases
Sending a local message	62	The time of sending a message to vicinity group members
The CAG maintenance server	63	The time of getting a CAG list
Collaboration component	656	The time of interacting with other CAGs

We investigate the effects of the number of CAGs on the average elapsed time over 20 runs of the above two group services. Fifty groups are first generated by simulation so that the number of group members in each domain is 4. For example, if the number of CAGs is four, then each group contains sixteen people dispersed evenly in four domains. The number of CAGs ranges from 2 to 10. Figure 9 shows that, the elapsed time of executing the two group services increases gradually with respect to the number of CAGs. The increase of the number of CAGs results in the increase of the number of network I/O events of the backbone network.

Thus, it takes more time waiting for the response packets from all CAGs. The long waiting time can be improved by decreasing the numbers of the interacting CAGs. For example, we can store all mobile users' locations into the CAG maintenance server in the future. Henceforth, when a CAG executes a service of searching all group members, it obtains the location context information from the CAG maintenance server instead of sending broadcast packets to all other CAGs.

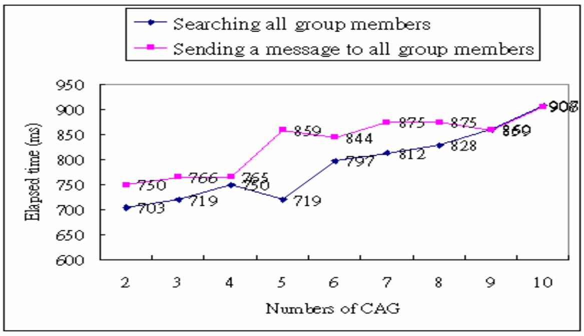

Fig. 9. The Elapsed Time of Group Services with Respect to Different number of CAGs

6 Conclusion and Future Work

We pointed out that past context-aware group communication researches could not support group communication across cells. We fixed the deficiencies by designing and implementing a context-aware group (CAG) communication system, which consists of one CAG maintenance server and several CAGs. The CAG system could track mobile users' current locations with a Wi-Fi AP, provide basic group communication services by utilizing and maintaining related context information, and help mobile users interact with their group members. The CAG system provides group services for the mobile users in the same or different localities, based on the peer-to-peer collaboration among CAGs. CAGs are connected by fixed networks. The use of the Wi-Fi AP enables a CAG to determine whether a mobile user is in its responsible domain. Our contexts are based on Dey and Abowd's definition. Contexts are also categorized into the primary and the secondary contexts, which make it easier to include more contexts in the future.

We expect to expand this research in the following directions:

1) Session migration: CAGs can be expanded with the capability of session migration, so that when a mobile user moved to another domain, s/he can automatically register with a new CAG. Thus, s/he could continue ongoing activities, such as receiving a file, within the new domain.
2) Group membership management: Group membership management was handled under simplistic assumption in this article. This research can be expanded to develop a mechanism of group membership management which dynamically divides the group into several parts, or merges several small groups into a large one.

Acknowledgment

This work was supported in part by Taiwan's National Science Council under Grant 94-2213-E-032-021.

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pp. 775-784, 2000.
[2] D. Bottazzi, A. Corradi and R. Montanari, "AGAPE: a location-aware group membership middleware for pervasive computing environments," in *Proc. of the 8th IEEE International Symposium on Computers and Communications*, Vol. 2, pp. 1185-1192, 2003.
[3] D. Bottazzi, A. Corradi and R. Montanari, "Context-Aware group communication in mobile ad-hoc networks," in *Proc. of the 2nd International Conference on Wired/Wireless Internet Communications*, pp. 38-47, 2004.
[4] P. J. Brown, J. D. Bovey and X. Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications*, Vol. 4(5), pp. 58-64, Oct. 1997.
[5] K. Cheverst, N. Davies, K. Mitchell and A. Friday, "Experiences of developing and deploying a context-aware tourist guide: the GUIDE project," in *Proc. of the 6th Annual Conference on Mobile Computing and Networking*, pp. 20-31, 2000.
[6] K. Cheverst, G. Smith, K. Mitchell, A. Friday and N. Davies, "The role of shared context in supporting cooperation between city visitors," *Computers and Graphics*, Vol. 25(4), pp. 555-562, 2001.
[7] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," Georgia Institute of Technology, College of Computing, Technical Report GIT-GVU-99-22, 1999.
[8] N. S. Ryan, J. Pascoe and D. R. Morse, "Enhanced reality fieldwork: the context-aware archaeological assistant," In Gaffney V., Van Leusen M. and Exxon S. Editors, *Computer Applications in Archaeology 1997*. British Archaeological Reports, Oxford, Oct. 1998.
[9] D. Saba and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *IEEE Computer*, Volume 36(3), pp. 25-31, 2003.
[10] M. Satyanarayanan, "Pervasive computing: vision and challenges", *IEEE Personal Communications*, Vol. 6(8), pp. 10-17, 2001.
[11] B. Schilit, N. Adams and R. Want, "Context-aware computing applications," in *Proc. of the 1st IEEE Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994.
[12] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, Vol. 8(5), pp. 33-32, 1994.
[13] R. Want, A. Hopper, V. Falcao and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, Volume 10(1), pp. 91-102, 1992.