GRECOM: Group Context Management
Middleware for Group Event-based Services

Chol-Song Chong and Nobuhiko Nishio

Graduate School of Science and Engineering, Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan
chol@ubi.cs.ritsumei.ac.jp, nishio@cs.ritsumei.ac.jp

Abstract. Although group context is an important element of user context, there are few services utilizing group context. A platform which provides group context will be essential when developing group-based services. GRECOM (Group Context Management Middleware for Group Event-based Services) works on each user’s mobile device and recognizes high-level group context by exchanging the packets. Based on the collected group context, GRECOM resolves various requests of services. GRECOM also prevents leakage of privacy information using the access control policy, which is altered according to user’s situation.

1 Introduction

People often form local groups with proximal people in daily life. So group context is an important element of context awareness. However, it is difficult to utilize group context as a trigger for service control (e.g. notify the escorting leader if a tour member gets lost during a tour). To achieve this, each user’s mobile device (node) has to recognize the user’s group context and provide it to services. In order to realize group-based service provisioning, we propose GRECOM (Group Context Management Middleware for Group Event-based Services). In the following, we explain about the services utilizing group context, requirements of the platform, and the architecture of proposed system.

2 Group-based Service

We define a group as a set of nodes which are proximal and sharing the same context. A group-based service is a service utilizing group context or group event, which means a specific change of group context. Group context includes membership, proximity, friendship, time together, coincidence of behavior or moving patterns, etc. Group contexts varies between nodes because each context is different from users (e.g. proximity, friend relationships). The following are simple examples of group-based services.

- Lost Member Discovery: If a member of a group (such as tourists) gets lost, other members quickly detect the occurrence and initiate support to reunite with the lost member.
– Context Label-based Resource Sharing: The member’s resources which are available from same group members change according to their context (e.g. tourists can share the pictures of the tour among them when they get together from free action time without specific actions).
– Group-based Advertising: Advertisements are delivered to the groups based on their situation (e.g. families who are walking in the outdoors receive a family coupon which is redeemable in a nearby restaurant).

3 Related Work

NearMe [1] Server computes a proximity of nodes from clients’ lists of Wi-Fi access points and signal strength. But such a centralized approach is not practical because of scalability and privacy problems. Meanwhile, several grouping protocols utilizing mobile ad-hoc network technologies have been proposed [2][3]. These autonomous protocols can provide the presence of group members, but providing only presence is insufficient for group-based services. People Near Me (PNM) [4] is a platform for collaborative services using P2P technologies. Application developers can build collaborative services by using the PNM API. But PNM assumes that the invitation of the user when starting a collaborative work. So, PNM is unsuitable for services which always need group context, as mentioned above. In addition, it is difficult to handle high-level group contexts.

4 Requirement

Considering the factors described above, we organized the requirements for the group context management platform as follows:

– Autonomous Grouping: Grouping mechanisms must not depend on specific infrastructure, such as Internet connectivity or positioning systems.
– High Level Group Context: This context is important for platforms to recognize high level group context, such as whether the user is walking alone or with friends in a crowd.
– Request Resolution: The API is essential for the application developers who describe service requests. For example, “How many people have been nearby more than 30 minutes?”, “Notify if the meeting members get together”.
– Adaptive Access Control: Because of privacy problems, platforms must control which context are transmitted to other nodes.

5 Group Context Management Middleware

GRECOM works on each node and resolve various request from services through its API. Group context is collected through exchanging the grouping packets with proximal nodes. Fig. 1 shows the architecture of GRECOM. GRECOM is composed of five modules and contains the data storage of group contexts,
private contexts and profiles, and trigger rules. We assume private contexts and profiles are collected from other resources of own node. Private contexts and profiles may include location, schedule, moving speed, communities that the user joins, contact lists (e.g. address books, friend lists for chat applications).

The Request Handler (RH) accepts the requests from services and returns the result according to request type. Requests are divided into queries of group context and operations (add, modify, delete) of group context. The former occurs when the service needs to reference group context. RH analyzes the query and obtain the required information. The latter occurs when specific group events and conditions are assigned. RH sends the event and conditions to the Trigger Manager (TM). TM generates the trigger rules and add to the rule table. Rules in the table are evaluated based on group context at uniform intervals. If the group event occurs, the service receives a notification from RH.

The Group Detector (GD) exchanges grouping packets with proximal nodes. At uniform intervals, GD broadcasts a hello packet (HP) containing the hash values (mention below) and contexts that are permitted to broadcast. Then, each node unicasts a context query packet (CQP) to other nodes in order to evaluate the relationship of proximal nodes. The nodes that received CRP send a context reply packet (CRP) as the answer of requested context based on their access control policy. The Context Manager (CM) estimates high-level group context based information from GD. For example, friendship evaluation is based on contact lists which, users maintain in daily life (shown in Fig.2). Each unique ID in contact lists is hashed in order to prevent the leakage of unique ID. Estimated context is stored in the group context table.

The Access Controller (AC) restricts the context which is transmitted based on a policy template. A policy template is an access control rule that a user can
configure arbitrarily. A policy template includes a situation label, list of context and permitted role (all users, friends, staffs, etc), which means access permission to each context. AC verifies the group and private contexts and profiles at uniform intervals through CM and selects the policy template which is best matched to current context (shown in Fig.3). If a policy template is altered, GD receives new context permissions.

6 Conclusion

In this paper, we propose GRECOM (Group Context Management Middleware for Group Event-based Services) in order to realize group-based service provisioning. GRECOM recognize high-level group contexts by exchanging the grouping packets and resolving the request of services through an API. In addition, GRECOM switches access control policies adaptively in order to prevent leakage of privacy information. Currently we are developing the prototype of GRECOM. We will verify the effectiveness of the proposed system in real field experiments. Moreover, we plan to establish a negotiation model which can alter access permissions according to other node’s access permissions in order to extract more high-level group contexts (e.g. node A tells the schedule information to node B only if node A receive the node B’s schedule information).

References