Coinonia: Privacy-Aware Meeting Scheduler based on Location-Based Services for Mobile Users

Hyoungtae Cho
hcho5@cs.umd.edu

Abstract
The location based service has been attracted more and more in recent years with the advance of wireless communication technologies. Some mobile applications are designed to provide users with their current location information shared by a group of people where a user might be reluctant to reveal the private information. The paper presents a meeting scheduling application, Coinonia to address a way of controlling activities in time to deal with the conflict between benefits of utilizing location information and risks of violating privacy in location tracking technologies. This privacy-aware application integrates Google Map views with the features of meeting scheduler, which enables meeting members to gather together with less effort by having them communicate efficiently.

Keywords: location-based service, mobile application, privacy, meeting scheduler

1. Introduction

A various set of new technologies in mobile devices and social networks has brought the rapid development of mobile applications. The new generation mobile phones called smartphones allow users more advancing computing powers to support the common features of computer or PDA as well as traditional phone-call functionalities. Among these features, location-based services have become more prevalent since a handheld device is one of tools used for tracking location and it prompts users for location information.

These mobile applications greatly enhance our quality of life with regards to providing users with needed information in the right place and at the right time. Such location systems, for examples, offer location-based services to give a driving direction to the nearest restaurants or gas stations for nearby residents or tourists. Location-based social networking services are also available to get in touch with not only friends and family but also interesting people of common interests. Moreover, they enable college students to notify campus policemen when they are in trouble. Despite the benefits of location-based technical supports, individual privacy is threatened by revealing the user's current location implying what he/she is doing to unwanted people or in undesirable time.

In this paper, we present a mobile application, *Coinonia* to share user location information with meeting members to efficiently get together. The application organizes meeting schedules inserted by users, shows the current locations of people who participate in the meeting during a certain time – *monitoring time*, and makes them communicate one another by sending instant messages. Once the meeting begins, the location services of this application is automatically terminated with a consent of meeting creator to protect location privacy. It also provides users with alarms personally set up in a mobile phone: 1) *late alarm* by computing the estimated arrival time to the meeting place and 2) *message alarm* on receiving a message from others.

The remainder of the paper is organized as follows. The next section briefly reviews the previous works related to our mobile application. In Section 3, we discuss motivations and background concepts to design the meeting scheduler. In Section 4, we present how to implement the application that meets design concepts and requirements. Finally, Section 5 summarizes our contributions and proposes future works.

2. Related Work

As a number of commercial mobile application have launched, the features of web-based scheduling programs such as Google Calendar[3] or HopStop[5] are integrated into mobile applications. Above all, however, the benefit of using these mobile applications is to utilize the user location information based on the static data inserted by users. Developments in mobile geographical positioning and location-based services have made it easier for mobile users to look up a physical location of interesting points or people nearby. For example, *TimeToGo*[2] is an application that combines personal calendars with public transit options and reminds users about upcoming events with departure alerts. This smart reminder system aggregates the components of diary, public transport timetable, and automatic alerts. Another location-based service[8] over the phone recommends popular social events around a user. Besides the personal use of location-based services, sharing location information like *Google Latitude*[4] or *Mologogo*[7] becomes popular as well.

Despite the benefits of mobile applications with location-based services, some researches[6, 11] have been conducted to examine privacy concerns in tracking location information. Among such applications to consider the privacy issue, Connecto[1] is an location sharing application that is concerned with privacy by allowing users to manually enter the name of location shared by a small social group to hide their actual location. Another example, Locaccino[10] provides users rich privacy controls set by group, time, and locations to resolve privacy issues. It also shows an interesting analsys that high entrophy locations, namely frequently visited places are less private by users.

Although these location-aware applications and their researches help to enhance location sharing privacy with more sophisticated settings, they have a limitation in using for a meeting scheduler to share location information with formal groups in most cases, not social groups including family members or close friends. While most of other mobile location services focus on an individual user's privacy preferences, *Coinonia* takes care of how the value of location information can be maximized in an agreeable manner among meeting members.

3. Application Design

We develop a mobile meeting scheduler, *Coinonia* to make our life easier, which provides users with location services of the meeting place and members as well as scheduling services in the perspective of information dynamics.

3.1. Motivation

Meeting is an important part of our daily lives since we continuously meet someone to communicate each other, to share information, or to make a decision. We write down a meeting schedule not to forget when, where, and with whom to meet. Moreover, until every meeting member arrives at a meeting, we contact one another to make sure if he/she will arrive on time or to leave a message to other members. Let us consider a scenario that one person wants to check the current locations of other members. It requires one-to-many conversations.

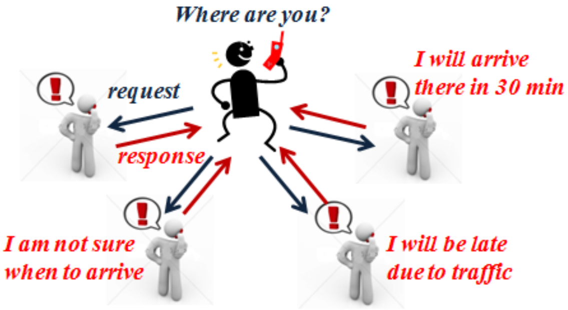

Where are you?
request
response
I will arrive there in 30 min
I am not sure when to arrive
I will be late due to traffic

Figure 1. One person needs *one-to-many* conversations to check other members' locations

On the other hand, an mobile application with location services over the smartphones makes it much easier, replacing one-to-many conversations for each person with one Map representation showing user geographical locations. Clearly, sharing location information among meeting members may help them to gather together.

Nonetheless, if a user's privacy cannot be preserved in using the location service after a meeting ends so that his/her current location is monitored by a group of people other than family members or close friends, the user will hesitate to use the application. Assigning a set of different privacy settings to meeting members would not be preferred since it reflects personal favor to affect relationships with group members. Regarding that logging off from the system is not easy to remember whenever the meeting is over, the monitoring functionality of this application is guaranteed to be automatically terminated when it is no longer needed.

3.2. Information-centric approach

In order to design a meeting scheduler, we take an *information-centric* view that explicitly consider the role of information and its dynamics in our system. Agrawala[9] claims that *every piece of information has value within a given context* related to its use and/or purpose and *the value of information depends on time*. In addition to his claim, the value of information may increase when it is given at right place and in right time for providing location information since a user's current location will also dynamically change in location-based services as time passes. Therefore, our application is designed to have a functionality of notifying whether or not a user will be late to a meeting based on estimated arrival time against meeting time.

Furthermore, his statements give us an insight to resolve the privacy issue in our information system. For instance, users can have more benefits sharing location information than costs of giving other their private location information to other meeting members within a certain time before the meeting. Once a meeting begins which means no location information is needed, the costs of sharing information become greater. As a result, the system should terminate location services at the end instead of remaining connected.

Why does the system have to take care of the termination instead of leaving it to a user's privilege? It's because the idle users tend to forget their log-off from the service as we can see the cases in school computers and because some reserved users would be anxious to become the first person to stop sharing personal information with others. *Coinonia* asks a meeting creator to terminate a service of meeting scheduler when the meeting begins, and makes the service completed one hour after the scheduled meeting time assuming that it would be rarely used anymore for a meeting purpose unless the meeting needs to be rescheduled.

3.3. System Architecture

In Figure 2, We explain system flows of *Coinonia* in two cases: a)when a meeting schedule is inserted by a meeting creator and b)when meeting members use it to meet together.

When a meeting creator creates a meeting schedule and *Coinonia*, its meeting information is inserted into the server through the HTTP server. In the meanwhile, the notification of creating the meeting is sent to each meeting member. As soon as member confirms the message, the application updates information about members with their confirmation, and starts to monitor a user's location for local late alarms based on his/her alarm setting. Within a monitoring time, the aggregated location information of monitored persons is shared by the meeting group, and they are allowed to send and/or receive messages through the application.

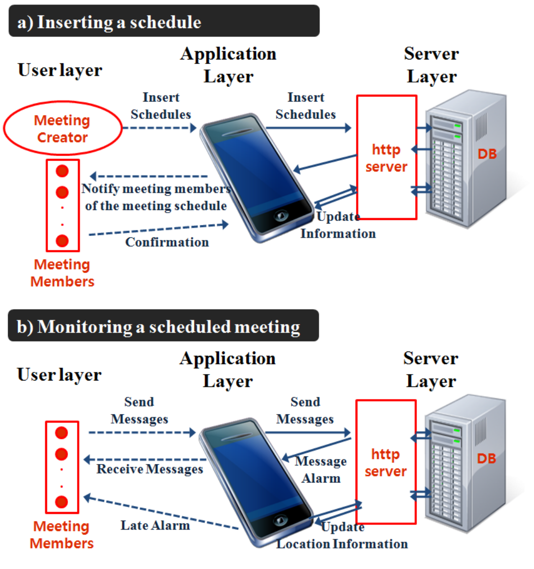


Figure 2. System Layout

4. Implementation

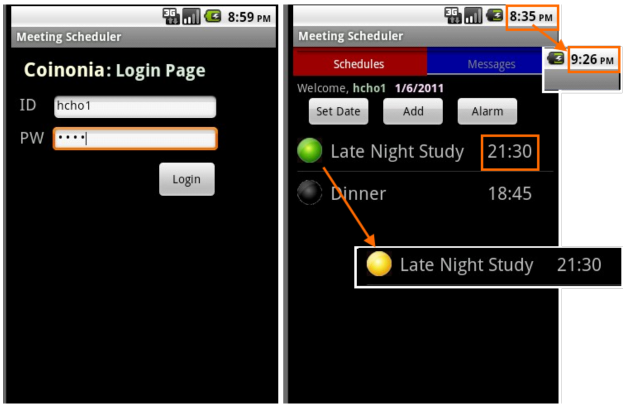
We develop and test our application on an emulator in Android 2.2 platform implemented in Java. It is an open source and enables Google Map features. To look up and store the data, we use a MySQL database server and communicate with it via the Apache HTTP server since our application cannot directly connect to MySQL through Android. *Coinonia* provides registered ID users with two main functions as a meeting scheduler: (1)Scheduling process and (2)Map Representation of user locations. They are accompanied with the instant messaging and meeting alarm.

4.1. Login Page

The first page of the application is *Login* page that requires a correct combination of *UserID* and *Password*. This process allows only registered users to access to the information system and to manipulate meeting schedules.

4.2. Main Page

When passing through the *Login* page, the *Main* Page in Figure 3 appears with two tabs – *Schedules* with a list of meetings that a user participates in and *Messages* with a list of received messages. Initially, a list of meetings scheduled on today's date comes out. Each row in the list consists of meeting title, time, and



Coinonia: Login Page

ID hcho1

PW ••••

Welcome, hcho1 1/6/2011

Late Night Study 21:30

Dinner 18:45

Late Night Study 21:30

Figure 3. *Login* page and Meeting List in *Main* page

a *green/yellow/red/black* circle indicating a possibility of arriving on time such as *safe/hurry/late/already past* respectively. In our example, a meeting schedule *Late Night Study* at *9:30pm* appears with a *green* circle since there was enough time left for the meeting comparing to its current time *8:35pm*, while time for another meeting schedule *Dinner* has already passed so that *Dinner* has a *black* circle. When it becomes *9:26pm* which is 4 minutes before the meeting, the color of circle was *yellow* since the user was near enough to arrive the meeting place (in about 11 minutes).

The details of color coding is shown as follows:

color	time range
green	$time_r > time_a + max(0.2 * time_a, 10)$
red	$time_r < time_a - max(0.2 * time_a, 10)$
yellow	between *green* and *red*
black	$time_r < 0$

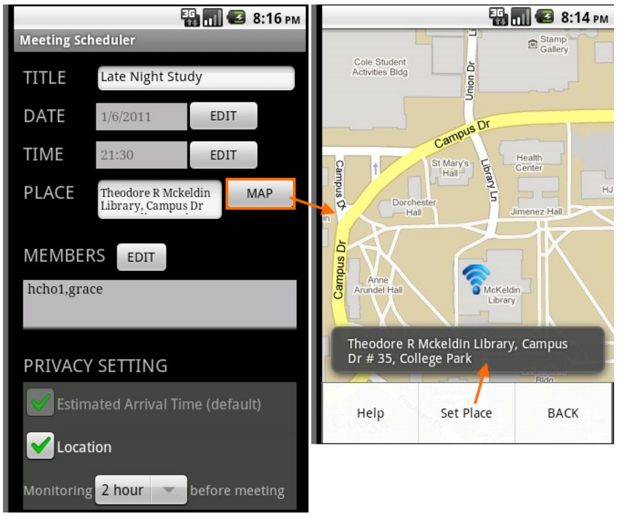
Table 1. Color coding for *late status*

In this table, $time_r$ is a remaining time for the meeting time and $time_a$ is an estimated arrival time from the current user location to the meeting place using Google APIs. All numbers are in minutes and rounded up. At present, computing arrival time is based on moving by car but it could be improved with more transit options.

The text message at the top frame includes the current user's ID and the current selected date for the meeting schedules. Among three buttons below, the user can navigate between dates using *Set Date* button and look through a list of meetings of the date. The rest of buttons are for creating a new meeting schedule and for alarm setting, and we will discuss them in the next sections(Section 4.3 and 4.6).

4.3. Create Page

The *Add* button in the *Main* page brings a user to the *Create* page where he or she can generate a new schedule. In this process the person becomes a creator of the meeting. He describes its details, decides which (both ID and without ID) members are participated, and sets up *monitoring time/privacy* settings. Once a meeting is created, each ID member in the list receives a notification and confirms to share his/her location information. Let us consider creating *Late Night Study* as an example.



TITLE Late Night Study

DATE 1/6/2011

TIME 21:30

PLACE Theodore R Mckeldin Library, Campus Dr

MEMBERS hcho1,grace

PRIVACY SETTING

☑ Estimated Arrival Time (default)

☑ Location

Monitoring 2 hour before meeting

Theodore R Mckeldin Library, Campus Dr # 35, College Park

Figure 4. *Create* Page

The meeting date and time is initially set to the current ones so that the application prevents the user from directly inserting date and time in a wrong format. Modification can be done only by clicking on *EDIT* button followed by date/time changes in a dialog box.

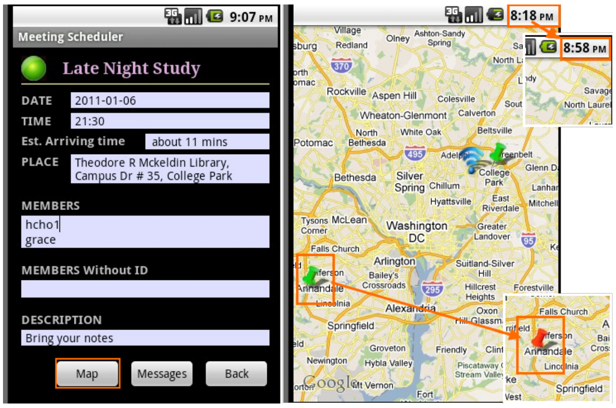
As for the meeting place, *Coinonia* allows users to set the location with pointing it out on the Google Map. Before setting it up, they are able to check its actual address by pressing on *A*(address) button. *Zoom in/out* functionalities and *Satellite views* are also available and all the corresponding function keys may be found in *Help* menu. Next, *Coinonia* obtains member information from users. Once *Members* section are filled out, the system recognizes which members are ID(registered) members or not and *Coinonia* will monitor locations of only ID members with confirmation in sharing their location information with other members.

In the privacy setting, each user is allowed to set up his/her privacy setting with regards to location and time. Whether or not to share *Location* and *Monitoring time* will be initially determined by a meeting creator. Location privacy setting has two levels: (1)*Estimated arrival time* from the current location to the meeting place and (2)the current *Location* where the user is located in. By default, *Estimated arrival time* must be checked out. How long a meeting member will be monitored by other members before the meeting begins is also selected on creating a meeting. When details of the new meeting are filled out, it will be inserted into the meeting list in *Main* page by *CREATE* button.

4.4. Details Page

If a user selects one of meeting list items in *Main* page, *Coinonia* enables a user to see its specific information such as *date*, *time* and *place* in the *Details* page. Figure 5 shows an example when the first meeting schedule *Late Night Study* is selected. The *Estimated Arrival time* is calculated for each member based on both his/her current location and the meeting place.

As we mentioned earlier, there are two groups of members – ID members with confirmation in using this location service and any other members – and the map representation with *Map* button represents only the location information of the first group marking with the color-coded pushpins. In this example, we can notice that a meeting member away from the meeting place(marked with a blue icon) was indicated by a red pushpin from a green one as the meeting time gets closer.


Meeting Scheduler

Late Night Study

DATE 2011-01-06
TIME 21:30
Est. Arriving time about 11 mins
PLACE Theodore R Mckeldin Library, Campus Dr # 35, College Park

MEMBERS
hcho1
grace

MEMBERS Without ID

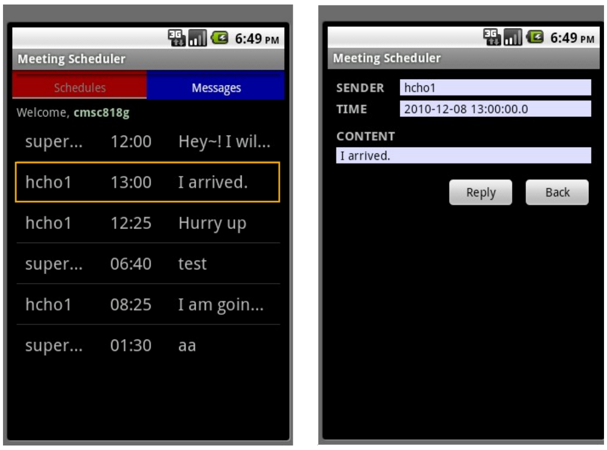
DESCRIPTION
Bring your notes

Figure 5. *Detail* Page and Map representations

If a pushpin is clicked, whom it represents and the user's brief message to other members, if any, are shown up. This map representation with location information saves time and effort to grasp the other members' location and situation on the way to the meeting place.

4.5. Message Communication

If users move onto *Message* tab at right top corner in *Main* page, they will see a list of received messages from others. The application provides a messenger function to efficiently communicate each other. A member can broadcast the current late status or leave a message to group members so that it may avoid one-to-many conversations as with map representation. Of course, one-to-one messages such as writing a message to a particular ID and replying it are possible. As with a meeting list, when one of message items is selected, its details come out (Figure 6) and a user can reply the message back to the sender.


Meeting Scheduler

Welcome, cmsc818g

super...	12:00	Hey~! I wil...
hcho1	13:00	I arrived.
hcho1	12:25	Hurry up
super...	06:40	test
hcho1	08:25	I am goin...
super...	01:30	aa

Meeting Scheduler

SENDER hcho1
TIME 2010-12-08 13:00:00.0
CONTENT
I arrived.

Figure 6. *Message* list and its Details page

For a particular meeting, if we click on *Messages* button in meeting *Details* page in Figure 5, we could write down a message to one of meeting member or all the members. The instant message function helps *Coinonia*'s efficient communication besides map representations.

4.6. Alarm Setting

Through the *Alarm* button in *Main* page, the alarm setting can be made personally. It allows users to have flexibility in receiving information for *late status* and notice of other members' messages. While the privacy setting is applied on the revelation of information to the entire meeting group, the alarm setting is locally applied on notification of his/her own late status and whether or not to receive messages.

5. Conclusion

In this paper, we present a mobile application, *Coinonia* to provide location based services preserving privacy. It is equipped with meeting scheduler functionalities and Google Map representation so that a group of meeting members easily gather together at a meeting place in terms of saving communication effort to find each other as well as notifying the user's current location. Time is a key variable to resolve a privacy issue by automatically terminating location information sharing among members as meeting starts.

At present, the further development of our mobile privacy preserving meeting scheduler focuses on two aspects: (1)testing *Coinonia* over smartphones in real environment settings to enhance its functionalities and (2)considering more transit options.

References

[1] Barkhuus, L., Brown, B., Bell, M., Sherwood, S., Hall, M., and Chalmers, M. From awareness to repartee: sharing location within social groups. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 497–506.

[2] Cameron, R., and Grantham, A. Timetogo: A mobile application for departure alerts based on google calendar, transit and gps data. In *UbiComp* (2009), ACM.

[3] Google. About google calendar. `http://www.google.com/googlecalendar/new.html`.

[4] Google. Google latitude mobile. `http://www.google.com/mobile/latitude/`.

[5] HopStop. Hopstop announces more mobile options! `http://www.hopstop.com/?action=newsletter&path=http://www.hopstop.com/img/news20101216.html`.

[6] Michael, K., McNamee, A., and Michael, M. The emerging ethics of humancentric gps tracking and monitoring. In *Proceedings of the International Conference on Mobile Business* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 34–.

[7] Mologogo. Mologogo. `http://www.mologogo.com/`.

[8] Quercia, D., Lathia, N., Calabrese, F., Lorenzo, G. D., and Crowcroft, J. Recommending Social Events from Mobile Phone Location Data. In *Proceedings of the* 10^{th} *IEEE International Conference on Data Mining (ICDM)* (Sidney, Australia, December 2010).

[9] Ronald, A. A., Larsen, R. L., and Szajda, D. Information dynamics: An information-centric approach to system design. In *Proceedings of the International Conference on Virtual Worlds and Simulation* (2000).

[10] Toch, E., Cranshaw, J., Drielsma, P. H., Tsai, J. Y., Kelley, P. G., Springfield, J., Cranor, L., Hong, J., and Sadeh, N. Empirical models of privacy in location sharing. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (New York, NY, USA, 2010), Ubicomp '10, ACM, pp. 129–138.

[11] Wang, J. L., and Loui, M. C. Privacy and ethical issues in location-based tracking systems. In *Proceedings of the 2009 IEEE International Symposium on Technology and Society* (Washington, DC, USA, 2009), ISTAS '09, IEEE Computer Society, pp. 1–4.