

# Enabling Group-Awareness through Context-Based Service Provisioning

Waltenegus Dargie  
Technical University of Dresden  
Chair of Computer Networks  
Faculty of Computer Science  
01062 Dresden  
waltenegus.dargie@tu-dresden.de

Alexander Schill  
Technical University of Dresden  
Chair of Computer Networks  
Faculty of Computer Science  
01062 Dresden  
alexander.schill@tu-dresden.de

## ABSTRACT

Platforms that support group-awareness and group interaction enable mobile users to have a rich knowledge about the whereabouts and activities of people with whom they work or live or have a shared interest. Context-based group-awareness enables members of a loosely-formed group to maintain the right balance between autonomy, privacy, and awareness. Previous work in this regard focuses on understanding the dynamics of groups and group members; on recognizing and sharing contexts; and on tackling scalability concerns. Little work has been done on the dynamic provision of service that are not foreseen when a group is formed. This aspect is rather an essential aspect, as it enables members to exercise autonomy and freedom. In this paper, we present a conceptual architecture consisting of core services and runtime services. The core services are the minimum amount of services required to create and maintain a group. The runtime services provide support that are specific to a group, and can be searched, installed, configured and integrated with the core services. The architecture has been implemented and deployed on a mobile device.

## Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems—*distributed applications*; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces—*asynchronous interaction, collaborative computing, theory and models, synchronous interaction*; H.4 [Information Systems Applications]: Miscellaneous

## General Terms

Conceptual architecture, modeling, prototyping

## Keywords

Context-awareness, group-awareness, group interaction, pervasive services

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CASEMANS '09, May 11, 2009, Nara, Japan  
Copyright 2009 ACM 978-1-60558-439-3 ...\$5.00.

## 1. INTRODUCTION

People undertake several activities in groups, at home, outdoors or in their offices and business environments. The level of dependency or relationship between the members varies from being closely tied until a set mission is accomplished to being casual, in which members can make independent decisions and come together as required, though members may want to remain aware of the whereabouts or activities of their peers. If the group is similar to the former type, the purpose of its existence is well defined; the contribution or the role of each individual member is well known; and a formal division of task is usually in place. Setting up this type of groups can be time consuming and elaborate; maintaining them, however, is not complex. On the other hand, in the second group type, the purpose of the group as well as the role of each member is diffused. Moreover, the degree of freedom of individual members to pursue unforeseen goals or goals that are not related to the existence of the group require flexible and robust interaction possibilities.

This paper focuses on the second type of group and aims to support context-based service provisioning in order to be able to provide members greater flexibility and autonomy. We will identify the core services that are required to create and maintain groups in which the members are free to pursue individual goals while maintaining an active awareness of the whereabouts and activities of their peers. First, we provide a conceptual architecture that is made up of five basic or core services. These services are useful to create groups, to seamlessly deploy and configure runtime services, to manage context; and provide a consistent user interface. Apart from the core services, there are also runtime services, which are group and member dependent. These services are deployed and configured at runtime. The architecture supports the core and runtime services to communicate with remote services to facilitate group mobility, context sharing and interaction.

In this paper, we discuss the motivation for such an architecture. We begin the paper by providing a scenario and identify key features of loosely-formed groups. This will be followed by the presentation of the conceptual architecture and its implementation.

In the remaining part of this paper is organized as follows: In section 2, we introduce a scenario both to motivate dynamic group-awareness and interaction and to describe the aspects of a spontaneously created group. In section 3, we provide a summary of related work. In section 4, we present the conceptual architecture and discuss its compo-

Time	Visit
08:30	Meeting in front of Summer hotel
09:00 - 11:00	Corcovado Hill
11:00 - 14:00	Drive to and Lunch at Botafogo Priara Shopping Centre
14:00 - 18:00	Drive to and sight seeing at Sugar Loaf

**Table 1: A hypothetical schedule**

nents as well as the interaction between them. In section 5, we present the implementation details and discuss our experiences. Finally, in section 6, we provide concluding remarks and future work.

## 2. SCENARIO

Traditionally, tourists have been traveling together to visit interesting places. There are different reasons why people prefer to travel in groups, but some of them are financial advantages, shared responsibilities and safety. Subsequently, travel agents setup groups based on convenience (mostly to the travel agents!) and shared interest; and outline tight schedules, deciding when and what the group should visit. These types of arrangements are known for their strict schedules and tight timetables, requiring tourists to accommodate partial loss of freedom. Moreover, unforeseen circumstances and opportunities are rarely accommodated. To highlight our point, consider the following scenario.

Five tourist couples from Germany meet through an online grouping service and decide to visit Rio de Janeiro. In Rio they want to stay close to each other and carry out some activities together. All of them are traveling with mobile devices that enable them to discover routes, sights, and public services. Moreover, the software platform on their mobile devices enables them to dynamically create and maintain groups; define access rights to the group members; exchange impromptu messages and multimedia contents; and seamlessly identify and download services that enable them to navigate through streets and complex shopping centres.

One of the couples will be staying in Leblon, in the Sun Hotel, and the rest will be staying in the Summer Hotel, in the Copacabana area. The tourists will democratically elect a leader to facilitate the group’s activity, but if the leader decides to engage in self-defined activities or not to join the group to a specific place, he either delegates someone or calls for a vote for a new election. Then the new leader will receive all the state information pertaining to the group activity. Among other things, the leader will be responsible to organise transportation and plan and execute visits. However, group members can be delegated to carry out subtasks in which case they have to synchronise their plan with the leader.

Let’s consider a hypothetical plan for Day 1 as listed in table 1

Some of the unforeseen events that should be expected and accommodated by the group look like as follows:

1. First couple (Sun hotel): The husband decides early in the morning to go for a jog and sets off, leaving a message to his wife. During the jog, he takes an extra time than he intends and on his way back to the hotel, he gets lost.

2. Second Couple: Since they have already been to the Corcovado Hill, they decide to spend the morning by sleeping a little longer and by having a relaxed lunch in the neighborhood. Shortly before lunch, however, the couple is already bored, and decides to join the rest of the group at Botafogo Priara Shopping Centre.
3. Third couple: They did not join the rest of the group to Sugar Loaf, and now at the Botafogo beach they are witnessing an exciting Jazz concert just kicking off and want to attract their group by sending pictures and clips to entice them.

These are common incidents among tourists. In (1), the concern is about time, or about keeping an appointment. Additional to the members’ unforeseen decisions and the consequences thereof, there are other factors that affect schedules such as the routes chosen between Summer Hotel and Sun Hotel; the congestion condition (which depends on the time of the day); the means of transportation (which in turn depends on the financial status of the tourists), etc. These factors usually frustrate users. Location-based services, most significantly, navigation related services, have been successful in the past ten or so years. Presently, a number of service providers associate information to users’ locations (such as points of interests, congestion information, etc.) and users plan travel routes accordingly. There are also services that estimate travel duration, but these are often coarse-grained estimations and not based on intelligent decisions that take congestion histories or driving habits into account.

In (2), the concern is similar to (1) in that the couple needs actual information regarding the whereabouts of their group and the estimated travel time, both of their own and the group to arrive at the appointed place. But there is also an implicit concern regarding dynamic and seamless integration of services that are spontaneously needed “on-the-fly”. For example, assuming that neither the couple nor the remaining members have never been to the shopping centre (which is a large place), they need to have an indoor location service and the transition from using an outdoor service to an indoor service should be seamless.

In (3), while the exchange of multimedia data is important, the question is how should it take place without causing much distraction to the users.

From the scenario above the interesting research questions can be formulated as follows: To enable the spontaneous creation of groups in which the members are free to pursue own goals and to facilitate interaction between the members:

- What kind of service support do members need to create and maintain the group?
- How can members effectively communicate with each other?
- How does service integration take place?

## 3. RELATED WORK

The psychological aspect of group formation and its technological implication is studied in [7]. A similar study based on Internet grouping (web-based geographic visualisation) and group merging strategy is studied in [5].

Bottazzi et al. [1] propose a group management middleware that facilitates dynamic and spontaneous group creation. The middleware uses users' location, attribute, preference and device access information to establish groups and to enable seamless resource management and sharing. The middleware is latter extended to support social networks of a larger scale [2].

Ganti et al. [8] propose the SenseWorld framework that enable virtual social networks to query physical objects which are associated with people and places. The sensor information is purposed to enrich members' knowledge of their group. SensorMap offers a geographical index into a virtual-physical social network with its own logical topology derived from social connections.

Ferscha et al. [6] propose a comprehensive framework for group interaction and localisation. The localisation framework provides location sensing that integrates different technologies and abstraction levels and resolutions.

The approach of Hallberg et al. [9] enables the dynamic management of advanced contact lists which can include presence and status information, synchronous and asynchronous multimedia communication tools, and methods for structuring social networks.

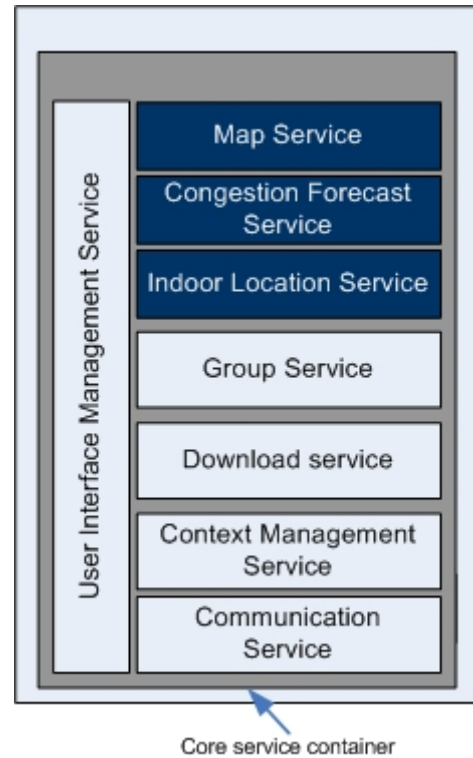
Gross and Prinz provide a concept for modelling shared contexts of physically distributed and independent users who dwell in cooperative environments. The concept includes the presentation of group-relevant contexts in unobtrusive manner.

Our approach builds upon some of the concepts that are presented and discussed in this section. The context recognition framework this paper employs is based on a previous work presented in [3] and [4]. While these approaches address particular aspects of group-awareness, mainly, the capturing and sharing of context information and efficient resource sharing, our approach focuses on providing a comprehensive tool to build and maintain groups; to enhance dynamic and seamless context and content sharing; and to dynamically identify, install and configure services (at runtime) that are very useful to enhance group-awareness and interaction.

#### 4. CONCEPTUAL ARCHITECTURE

We would like to address two important concerns in our conceptual architecture: mobility and group-awareness. The first concern is to enable users to freely move and maintain access to information that is useful both to their mobility as well as to their knowledge of their group members (such as their whereabouts and activities). The second concern is to enrich the users' ability to interact with their group members through a diversity of ways. This includes enabling users to dynamically identify, locate, configure and interact with services in an impromptu fashion. Our aim to address both concerns is through context-awareness and dynamic integration of pervasive services.

The conceptual architecture we propose consists of two basic classes of services. The services in the first class are called core services. These services are light-weight services and will be available on all mobile devices that should support group-awareness. The main envelope that contains these core services can be downloaded at runtime and is capable of configuring itself, prompting users to create groups and interact with them. Typically, these services enable group



**Figure 1: A conceptual architecture for group-based interaction**

formation, dynamic service configuration, remote communication, and user interface management.

The second classes of applications are those which can be dynamically searched, downloaded, and configured, as needed. Therefore, their scope and usefulness depends on the context of their users. We have defined and implemented some of these services, but it is possible to define new services that have not been foreseen so far. Figure 1 shows the basic structure of the architecture. The services in gray are the core services while the blue services are the ones that are context dependent.

In the following subsections, these services are described in more detail.

##### 4.1 The Core Service Container

The core service container is responsible to manage the life cycle of all services that support group-awareness, i.e., it loads, configures, starts, stops and removes a service. Moreover, it exposes basic controlling functionalities to the user so that the user maintains full control of the services that are running on his/her mobile device. When it first starts, the core service container's main task is to start and configure the core services.

##### 4.2 Communication Service

The communication service provides basic communication functionalities to all other services which require remote communications and inter-process communications, i.e, communication between independent services that are running on the same mobile device. It does so by interfacing the higher-level services with the lower-level networking services,

thus, managing the complexities of establishing communication links and by abstracting the change in the underlying communication infrastructures. All the other services communicate with it via messages, leaving the concern of communication details to it.

### 4.3 Download Service

The download service is responsible for searching, selecting, downloading, installing and configuring a service. For local services that are not yet active, the download service is responsible for loading and starting them as needed and handover control to the core service container for life cycle management. Additionally, it is responsible for updating a service. On the other hand, it is a gateway for other mobile devices that are searching for a new service for a given task.

Identifying services in a dynamically, mobile environment is not trivial. The first challenge is to localize the search. As users leave desktop environments and move into resource constrained, mobile environments, both the search time and the search space should be limited. Secondly, even when the right services are identified, the downloading bandwidth is narrow and, possibly, expensive. For these reasons, we limit the search space of the downloading service within the devices of the group members<sup>1</sup>. An associated problem with dynamic peer-to-peer service discovery (and therefore the justification for limiting the search space) is the simplicity of the service description that can be supported. In our implementation, we used a simple service schema with a predefined structure to define the type of service that can be provided, the interfaces that can be invoked, and the input and output data types of the services.

### 4.4 Context Management Service

Context-awareness is the core aspect of group-awareness. Understanding the context of users and the devices and places they are associated with them enables members to know what is taking place in the group while maintaining independence such as freedom of activity and movement. Moreover, context-awareness is useful to determine and get the services users require. Users supply context information to all local services and to the group members through context providers, each of which abstracts the acquisition of a context from a sensor, an application, an infrastructure, or a device. The context management service is responsible for configuring and managing the context providers. The context providers are integrated in the platform as runtime services. The context management service enables users and services to query and subscribe to context providers and context events. When users are interested to subscribe or query remote providers which do not belong to them, it is done by establishing a remote request to the remote context management service. The remote context management service refers to the local privacy policy before granting access to context providers.

### 4.5 Group Management Service

The group management service is responsible for creating and managing a group. The management aspect includes, defining a set of tasks that will be carried out by the group members and associating these tasks with all the relevant

<sup>1</sup>We do not consider it feasible to support web services in mobile peer-to-peer environments due to the slow response of these servers and the associated communication cost.

context information (location, ID, time, etc.). Additionally, it enables members to specify access rights to their peers to personal services (such as context sources), files and file folders. For example, members can be granted the right to subscribe to multimedia contents, but due to bandwidth and energy constraints, they may restrict the content types (picture only, for instance) as well as the access frequency (every 30 minutes, etc.). This applies also to the context sources, as the frequency of context information delivery determines the amount of local resources that can be consumed.

### 4.6 User Interface Management Service

The user interface management service is responsible for presenting to the user all the services that are running and require a user interface as unified services. It does this by presenting a consistent and unified view. Each service describes its user interface using an interface definition schema and deliver this description to the user interface management service which in turn creates a (graphic) user interface and connects it with the messaging service that channels the input and output data of the service.

The integrated user interface environment enables services efficiently use a shared space. Moreover, it enables service developers to declaratively express their input and output requirements, leaving the implementation details to the user interface management service. The user interface management service is also responsible for orchestrating input and output relationships such as managing sequential and parallel inputs and outputs, based on the control information of the interface description that is acquired from each individual service. Because of the complexities of orchestration and control, the user management service manages one service at a time and does not deal with composite services.

### 4.7 Interaction between Services

Figure 2 shows how the different local and remote services interact with each other. The two separate boxes represent two devices owned by two users. The left device is configured with the six core services<sup>2</sup> and two runtime services, i.e., the context provider and the map service. Likewise, the right device is configured with two runtime services (the multimedia sharing service and the context provider). The configuration demonstrates a scenario in which the left side device subscribes to the multimedia service and the context management service of the right side device. This way, for example, the left side user remains informed of the whereabouts of the second user as well as the places she/he visits, provided that the multimedia content in the multimedia file folder refers to the present location of the user.

One way to do this is abstracting the user's camera with the multimedia sharing service which then stores pictures and video clips with a label or a name associated with the location from which they are taken. This can easily be achieved if the multimedia service subscribes to the context management and obtains location and time information whenever the user takes a picture of a video clip.

The multimedia sharing service automates access policies by associating context information with the user's request. For example, the user can provide GPS coordinates and time information to the multimedia sharing service to grant or deny access to specific multimedia content.

<sup>2</sup>The download service is not displayed here for convenience.

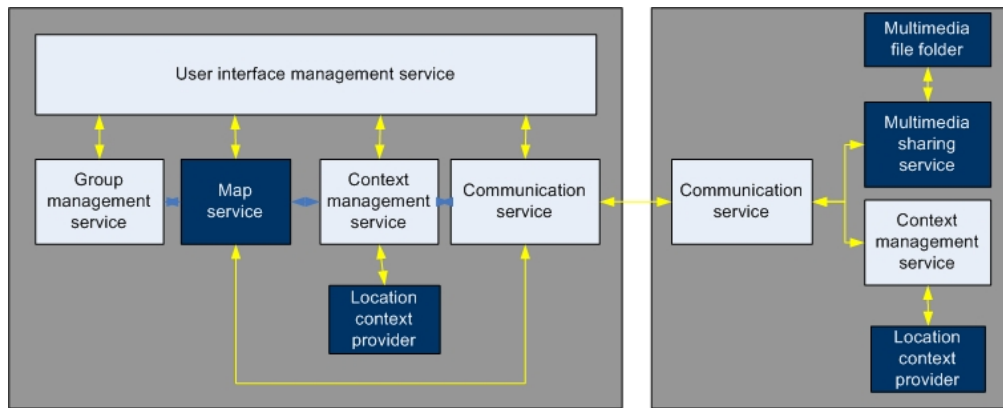


Figure 2: Interaction of local and remote services

Technology	Specifics
Java runtime environment	IBM J9VM (ive 2.2) JCL personal profile 1.0 XML processor (DOM) JNI class to execute IE Mobile
OSGi	Equinox 3.2.1 HTTP and servlet interfaces
Hardware	Pocket PC ARM processor Bluetooth and WLAN support GPS receiver with Bluetooth access 30 M available memory
Software	Window mobile 5.0

Table 2: Summary of the technology used to implement the conceptual architecture

The multimedia context obtained from the remote user can be dynamically obtained by the map service and displayed to the user in different ways, depending on the user's context and preference. For example, it can be displayed in a map that shows streets and shopping centres or it can be displayed by the user interface service as a stand alone image.

An important aspect of the group-awareness platform is that services are loaded, started and stopped as they are needed. In our implementation, services are loaded when a subscription or query request that refers to them is obtained from a local or remote user and only when the request is approved by the group management service.

## 5. IMPLEMENTATION

We implemented the conceptual architecture using the Java runtime environment (IBM J9VM) and the OSGi platform on pocket PCs. The devices we used had ARM processors and Bluetooth and wireless LAN capabilities. The two network interfaces were used to exchange multimedia content and context information. Windows Mobile 5.0 was installed on the mobile devices. A summary of the technologies we used and their brief description is given in table ??.

The prototype includes all the core services and additional runtime services, including a map service (that integrates Google Map), a calendar service, and a location provider.

All services are realized as OSGi service bundles, and therefore, can be downloaded, started, stopped and removed from memory at runtime. The group management service enables users to define and manage multiple groups. It uses the calendar service to create and execute time-dependent group activities. The calendar service has additional service as a context source because user's can inter into it activities such as meetings.

The user interface management service creates a unified user interface by integrating the group service and the map service. While the group service enable users to manage and interact with their group members, the map service displays the locations of members along with additional multimedia flags. The detail of the flags is configuration dependent, and may contain a link to additional services, such as a media player.

Locally, the services communicate with each other by exchanging XML messages. HTTP was used for remote communication. The core service container uses the OSGi basic life cycle management functionalities to download, install, configure and start OSGi bundles (services). At present, the scope of peer-to-peer service discovery is within the members of the same group. As a result, the download service begins a search request after members have been invited and after they have confirmed the acceptance of the invitation.

The communication service uses the WLAN interface to support peer-to-peer communication and the Bluetooth interface to connect the GPS with the location context provider. The Bluetooth interface is also used to share multimedia content, but this is rather very slow and often unreliable.

2 shows an instance of a single configuration for the group-aware service. In this setting, two members of a group are interested to share context information and multimedia data. The right side user makes multimedia data accessible through the multimedia sharing service, which in turn accesses the multimedia folder. The multimedia folder can be attached with a multimedia source, such as a camera, and multimedia content can be automatically stored in this folder, as the content is made available. The multimedia sharing service can be configured to determine the type and size of the shared content as well as the frequency of the remote access.

On the other side, the subscribing member employs the communication service to receive the multimedia content and the context information. Once the data are received,

they will be provided to the map service which filters the data such that they can be presented to the user. Logically, the map service, on one side, and the multimedia sharing service and the context management service, on the other side, should agree on the type of the data as well as the frequency of their delivery.

## 6. CONCLUSION

There is a substantial work on group-based interactions. Most of these use the Internet as the underlying infrastructure to create groups based on preferences and to exchange multimedia contents and to maintain social networks. Several of these applications have been useful and effective. On the other hand, there are not that many services and infrastructures for supporting group-awareness and group-based interactions. Despite the pervasive availability of communication infrastructures and location based services, there is a conspicuous lack of coordination between them. As a result, users usually spend considerable time to manage their activities and group interests.

We provided a conceptual architecture for supporting group-awareness. The architecture consists of some basic services that enable peer-to-peer communication, dynamic service download and configuration, context management and group management. The architecture also enables the dynamic integration of runtime services which can be downloaded, configured, started, stopped or removed from memory according to the contexts of the users, such as their immediate goal, activities and whereabouts.

While group-awareness implies that users are distributed in physical places and engaged in their own activities while maintaining sufficient knowledge of the activities and whereabouts of their group members, but it also means that members come together and accomplish a task or engage in common activities, possibly in an impromptu fashion. In this case, the group management service should facilitate the coming together of members. One way to achieve this is by providing users with context-dependent information about transportation alternatives, including cost and travel durations. The services that are available to date are inadequate in that they do not fully utilize context information and often do not easily integrate with other services and platforms.

By contrast, our conceptual architecture enables the dynamic integration of services. We achieve this by enabling services to expose their lifetime management functionalities to the group management service and by supporting context-based service download and configuration. The download service searches relevant services (where relevance depends on the user's interest) within the devices of the group members and downloads and configures them at runtime. These services, in turn, subscribe to available context providers. If there are no context providers, they too, are downloaded and configured at runtime.

## 7. REFERENCES

- [1] D. Bottazzi, A. Corradi, and R. Montanari. A context-aware group management middleware to support resource sharing in manet environments. In *MDM '05: Proceedings of the 6th international conference on Mobile data management*, pages 147–151, New York, NY, USA, 2005. ACM.
- [2] D. Bottazzi, R. Montanari, and A. Toninelli. Context-aware middleware for anytime, anywhere social networks. *IEEE Intelligent Systems*, 22(5):23–32, 2007.
- [3] W. Dargie. The role of probabilistic schemes in multisensor context-awareness. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 27–32, Washington, DC, USA, 2007. IEEE Computer Society.
- [4] W. Dargie and T. Tersch. Recognition of complex settings by aggregating atomic scenes. *IEEE Intelligent Systems*, 23(5):58–65, 2008.
- [5] M. Elias, J. Elson, D. Fisher, and J. Howell. Do i live in a flood basin?: synthesizing ten thousand maps. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 255–264, New York, NY, USA, 2008. ACM.
- [6] A. Ferscha, C. Holzmann, and S. Oppl. Context awareness for group interaction support. In *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 88–97, New York, NY, USA, 2004. ACM.
- [7] D. Fisher. Using egocentric networks to understand communication. *IEEE Internet Computing*, 9(5):20–28, 2005.
- [8] R. K. Ganti, Y.-E. Tsai, and T. F. Abdelzaher. Senseworld: Towards cyber-physical social networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 563–564, Washington, DC, USA, 2008. IEEE Computer Society.
- [9] J. Hallberg, M. B. Norberg, J. Kristiansson, K. Synnes, and C. Nugent. Creating dynamic groups using context-awareness. In *MUM '07: Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*, pages 42–49, New York, NY, USA, 2007. ACM.