A CONCEPTUAL MODEL FOR SUPPORTING SOCIAL GROUPS WITH UBIQUITOUS COMPUTING

by

John Bodily

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master Of Science

ARIZONA STATE UNIVERSITY

May 2006

A CONCEPTUAL MODEL FOR SUPPORTING SOCIAL GROUPS WITH UBIQUITOUS COMPUTING

by

John Bodily

has been approved

April 2006

APPROVED:

______________________________, Chair

____

____

Supervisory Committee

ACCEPTED:

____

Department Chair

____

Dean, Division of Graduate Studies

ABSTRACT

This thesis details a model designed around the cognitive analysis framework of Activity Theory which addresses the social group member's perception of how computing devices should support them in a Ubiquitous Computing environment. From the conceptual model a prototype ephemeral group membership management system and a sample Group Chat application for use over ad-hoc networks were developed. Findings from usability testing the prototype indicated: In order to reduce interruption users desire active feedback about how context-awareness is used to automate group interactions. The existence of a group interaction for a situation is a context that can be used by applications to support groups. Users desire functionality that leverages contextual properties specific to the group as a whole, and to a user's group memberships.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1. Motivation

The primary goal and contribution of this research is to provide a better understanding of how groups should be supported in a Ubiquitous Computing(Ubicomp) environment. In order to achieve wide spread acceptance of Ubicomp technologies computing should embrace the end-user not only on an individual basis, but should provide support for the social groups that they are a member of. Furthermore this must be accomplished in a manner that does not distract them while they are collaborating. To do this a means of leveraging context-aware computing using contexts about social situations that may not be detectable using sensors, but are important to the user must be found.

In his seminal paper The Computer for the 21st Century [23] Mark Weiser describes his vision of how the next generation of computing technologies will become such an integrated part of every day life that it will seemingly disappear. In order to fulfill this vision future-computing technology must be integrated into the home and work place in a way that minimizes the amount of cognitive load placed on the user whilst using those technologies. When using a computer the user must focus on the task at hand and not the constraints placed upon how they carry out the task based on computing infrastructure constraints.

In order to accomplish this level of usability computing must support the user's intent proactively, and serve context relevant content. Context-aware computing [10] is seen as a major component of the ubiquitous computing movement. By allowing devices to interface with both the end-user and the environment, cues can be obtained by collecting data from inputs that were previously unused. It is hoped that by being aware of information relevant to the state of the environment and user such as location, noise, light, identity of others nearby, and mood that new levels of usability can be achieved to support the end-user.

Frequently a user will be trying to accomplish something with a group of individuals. Thus it is natural to assume that the intent of the group will need to be supported in order to achieve the vision of "calm-technology". Indeed a group of individuals is a source of contextual information. The purpose of a group, and the roles of it's members often dictates what behaviors are acceptable in a given situation. These contexts known, as social contexts cannot be detected themselves by sensors, and thus another means must be used to represent these contexts, detect them and leverage their use.

2. Contributions

In order to address the motivations described in section 1 a conceptual group model was developed and prototype software was built to implement a collaborative system using the model. As a result the following concepts were elucidated from usability testing:

1. Keeping the user informed through active feedback about context and how it is used by an application to support group interactions is part of minimizing distraction.
2. The existence of social group collaborative sessions is a context that applications can leverage to support social groups.
3. The formal and informal group memberships of a user can be used by context-aware software to support social group interaction.

In the case of software that is used to support groups through context-awareness it is necessary to keep the user properly informed about system state from the contexts being used. Interactions with groups are far more complicated even for people to handle without software assistance. As such the user is more likely to be distracted and frustrated when the software does not prompt them for suggestions and authorization to take action based on a situation. Usability testing confirmed this when users felt a loss of control due to automated support for group chats proactively being shut down due to location changes and they were not properly informed as to why support was dropped. A more obvious example for when it is pivital to ensure proper feedback to the user involves decisions about membership.

Knowledge of groups that are active or will be active in a location or for a given situation can be used as a means to aid social groups and users. It can help users find groups and other users that they may wish to interact with, and aid groups in checking the availability of resources including rooms, and computers. This is a group-centric means of supporting context-aware proactive functionality from the perspective that it uses social contexts about the group as a whole. Testing indicates that in some cases users want to work with groups based on properties of the group as a whole, but in other cases want support based on properties of individual members.

Memberships of a user are an important user-centric context that can be used once made accessible to context-aware software in order to form group sessions. Knowledge of a users role as a student in a specific class from a schedule can be used to create a collaborative session for people interested in talking about the class. Other information about what the user may be interested in such as hobbies can be used to suggest collaborations by forming on the fly informal group memberships to introduce people to each other.

3. Background

The next chapter is dedicated to discussing existing research that was relevant to the body of work contained in this thesis. An overview of the concepts pertaining to research in the area of ubiquitous computing is covered first. Next, more details about efforts in the area of context-aware computing (a sub-area of Ubiquitous Computing) will be given. Efforts made in Ubiquitous Computing research to support groups of users working together through location awareness and using chat applications is presented, as well as the use of automated user profiling to create computer supported social functions.

Portions of this research were inspired due to the reviewing of three Psychological models used in the CSCW community for studying human cognition. The theories of Situated Action, Distributed Cognition, and Activity Theory are briefly described, along with a statement of why and how Activity Theory influenced the development of some of the concepts presented by this work. The chapter is ended by discussing the basic procedures employed when conducting usability testing to assess computer system interfaces and workflow.

4. A Model to Support Social Groups

The primary contribution of this research was the development of a conceptual model for supporting groups in a Ubiquitous Computing environment that was presented in [21]. The third chapter reviews the motivation, challenges, concepts and construction of a model for supporting social groups, known as the Social Group Model. The Social Group Model is developed around the definition of a group as used by Sociologists in [12]: "A number of individuals, defined by formal or informal criteria of membership who share a feeling of unity or are bound together in relatively stable patterns of interaction". Groups as described by this definition are termed Social Groups by this research, and focus is placed on how a model must envelop the concepts of group membership, and context-predictable repetitive interactions.

A model used to design and develop software for the purpose of supporting social groups must contain functionality to construct memberships. Memberships can be both formal such as a class, and more informal and ad-hoc such as striking up conversation with people nearby. Groups generally perform repetitive interactions such as a software development team conducting weekly meetings to review progress and present demonstrations. Friends on the other hand interact routinely when they speak with one-another based on knowledge that the other is nearby. Context-awareness is seen as a means to provide proactive computerized support for group interactions.

In this research context is not limited to include only data measurements that are taken by sensors, but more socially defined and understood contexts. Examples of such contexts that affect group interaction are social roles, skills, and the interests of individuals in the group. These contexts are defined as "Social Contexts". The model explores a means of representing, and leveraging Social Context in user-defined profiles in order to determine when a group will start and end an interaction.

The Social Group Model is intended to be used as a conceptual building block for constructing software systems to support groups using social contexts through context-aware computing. Context-aware computing using social contexts is intended to be used as a means to support a perception that several networked devices are supporting the purposes for a group collaboration that the user is currently or will be involved with. If this is achieved then the user will feel that the computing software supports their understanding of group activity.

5. Prototype Group Collaboration System

In order to establish the feasibility of using the social group model as a cornerstone for developing group collaboration systems a prototype system was constructed. The system was developed using Dell Pocket PCs as the platform device to be used in group collaborations.

The system consisted of system software and applications to support creating groups to interact with through chatting over ad-hoc networks. The following list contains brief descriptions of the software developed for this research:

1. **Situation Aware Ephemeral Group (SAEG) Service -** The SAEG uses context-awareness to form group views that allow an application to send a message to several devices at once using a group identifier instead of individual I.P. addresses.

2. **User Profile Distribution(UPD) -** Used to share contexts relevant to the interests, roles, and preferences of the end-user with other users in the area.

3. **Group Manager -** A management application for configuring groups that will be activated by the SAEG. The contexts that activate groups are specified here, and groups can be shared with other nearby users with this application.

4. **Group Chat -** A chat application for exchanging messages, and files with groups of users. The Group Chat uses the group created and managed by the Group Manager to support context activated group interactions.

A primary goal of constructing this prototype was to perform usability testing. Testing was conducted in order to determine usability problems specific to a system used to support groups in Ubiquitous computing environments. Reviewing usability problems was used as a vehicle to discover concepts that can be factored into systems and applications that support group collaboration using social contexts.

6. Usability Testing Results

This chapter is dedicated to overviewing the results of usability testing that was performed for this research. The background of users that participated in testing is documented, as well as an overview of the various tasks that were completed. Results are summarized next.

The average time it took users to learn how to create and join in a group with another user with the group manager was found to be an average of 10 minutes and 36 seconds. The first usability issue found dealing with the importance of providing proper feedback to the user in terms of how an application uses context is delved into next.

The second major usability problem was found because users indicated the most difficult portion of working with groups to be the final step of forming the group with another user. Through observation this was also the most time-consuming step of forming a group. The usefullness of using the social context of group existence for a given situation is explored, and is seen as a means to improve usability for forming groups. Also, the use of social contexts about the user such as interest, and schedule as a means for proactively finding or forming groups to aid in usability and fostering in-person interaction is explored. The concepts for constructing a service that comprises functionality to address the major usability problems as found from our tests is detailed and proposed as future work.

CHAPTER 2

Related Work

Ongoing and previously conducted research that is related to the body of work contained in this thesis is presented in this chapter. First, work in the area of Context-Aware computing is reviewed. Then the concepts of Ubiquitous Computing research will be covered. Next specific research in the area of Ubiquitous Computing that deals with supporting groups of users and social functions by profiling users will be covered. Theories on human cognition that influenced the research in this thesis are presented next. Finally the general guidelines for usability testing that were followed by this research to evaluate software are included.

1. Context-Aware Computing

Context-aware computing creates computing systems that interface with the end-user, and the environment within which they are used [11]. Making applications context-aware is being explored as a means to produce applications that are easier to use, and have new previously un-offered features. Useful contexts that have been successfully leveraged for use by computing applications are identity, and location. An example of a context-aware application that uses the contexts of location and identity is a call forwarding application for an office building, which uses the location (i.e. room) and identity of a person to forward a call to a phone that is accessible to them [22]. This application relies on an infrastructure that manages the location of all users in the office building in a query able central repository. Location and identity in this case are managed through an active badge system. An active badge is badge with an IR transmitter that is carried by a person which transmits a unique identifier to nearby IR sensors that are embedded throughout the rooms in an office space. While this application only showcases location and identity there are a broad set of other contexts that can be captured including noise, light, and motion [17]. However, in general the use of these contexts has either been left unexplored or their usefulness is not well understood.

Context-aware applications adapt their behavior in order to better meet the needs of the end-user in their current environment, while trying to make these adaptations as unobtrusive as possible. Applications such as the CyberGuide [2] from Georgia Tech use the context of location to provide relevant and useful information to tourist about nearby attractions, or displays inside a museum. Portable Help Desk (PHD) [4] developed at Carnegie Mellon University uses both spatial and temporal context to provide several services to students when they are on campus. This includes map services that display nearby available resources which can be computing and non-computing related, such as nearby printers, and events on campus. PHD can also be used to locate other individuals on campus whom may be a friend or a class team member.

2. Ubiquitous Computing

In his seminal paper "The Computer for the Twenty-First Century" [23], Mark Weiser discussed his vision of computing as not only occurring through interactions with a desktop computer, but anytime/anyplace with a myriad of computing technologies that would be embedded throughout the environments within which we live. This notion was developed based on the observation that the trend in computing advances has caused the ratio of computer to user to go from 1:n, to 1:1, and very soon it is anticipated that this ratio will become n:1. Indeed this is quickly becoming the case as the result of research in distributed systems and the wide spread adoption of mobile computing devices such as PDAs, and cell-phones. Since the concept of ubiquitous computing was first published several notable research projects have been undertaken including project Aura at CMU [4], and Easy living at Microsoft Research [15].

Researchers at Carnegie Melon university have developed the Aura system to explore ubiquitous computing on the college campus. Aura seeks to minimize distraction felt by users by leveraging context-aware proactivity and task migration. Two context-aware applications built using Aura are the Portable Help Desk(PHD) and Idealink shared whiteboard. PHD as previously described in section 1 is used on a mobile device such as a PDA and allows access to campus events, the location of classmates, and aids the user in finding suitable computing resources such as printers with low queues. Idealink is a shared whiteboard used to support group brainstorming and collaborative drawing. Idealink uses PHD to access user schedule information and determine whom will take part in a collaborative session.

Distraction can be caused to a mobile computing user due to differing levels of infrastructure resources throughout an environment. Aura can migrate a computing task started in one location to another as the user uses computers in different areas. Aura tracks information about computing resources in specific locations and seeks to maximize the performance and utility of computing tasks when they are migrated from location to location. This is done using resource profiles kept on tasks which are shared through a distributed file space.

EasyLiving by Microsoft Research mainly focuses on ubiquitous computing for use in the home. Similar to Aura computing sessions follow users around an area based on location-awareness. A specific example involves migrating a session from a small computer screen to a larger screen when a user moves from a desk to a couch. User preferences for lighting, and music are also used based on identity. When a user enters an area and no-one else is present it will be configured to their lighting and music tastes.

3. Support for Groups and Social Functions in Ubicomp

An important aspect that must be addressed to meet the Ubiquitous computing challenge is non-intrusive support of not only individual users but also groups of users [6]. Although computer support for groups is not a new area of research, the emergence of Ubicomp techniques are seen by some researchers as a means to improve how we work, and play as groups. This section describes systems that use location to support facilitating or tracking group interaction, and the use of context-aware chat or, instant messaging as a means to interact with groups.

3.1. Location Used to Support Groups. Some work in the field of context-aware computing for the support of groups has been performed. In [24] the context of group identity and location is used to open and close computerized documents that are associated with the work a group is performing. In the PEPYS project [20] we see another use of the active badge system to aggregate and provide information about the contexts of location and identity to a worker at the EUROPARC research facility. The PEPYS system tracks the location of the worker throughout the day, and the identities of other workers that they were collocated with at a given location for a given time interval. At the end of the day an active badge user will receive an email summarizing their day at work. The system tries to summarize the basic activities that the person performed throughout the day, and with whom the activities were performed. In particular it tries to categorize interactions between PARC workers into one of the following occurrences such as: in office, attended event, attended meeting, etc.

3.2. Context Aware Chat and Instant Messenging. Context-awareness has been blended with instant messenger and chat applications in the past by the Hubbub [9] project at AT&T, and by the ConChat [14] application developed at University of Illinois at Urbana-Champaign. The Hubbub project uses the context of sound, to foster opportunistic interactions with users on the buddy list by playing a unique sound for each user when they become available. ConChat uses context-awareness to aid parties in determining the overall context in which an on-line chat is occurring, by displaying the context present in the room where a remote user is chatting from. Neither Hubbub, or ConChat support messaging without support from a centralized server. An Instant Messenger is used as a primary interface for supporting group interaction in a Ubicomp environment in [1]. Similar to our notion of Social Context they introduce the term Group-context. In both their definition and ours the contexts used to support groups are an aggregate of several other base contexts, and once again location is seen as one of the most pivotal.

3.3. Social Support and User Profiles. In [5] users are statistically profiled for interests as they move around a conference based on whom they spend time near, and what displays they linger next to in order to suggest new or similar topics they may find useful. Current acquaintances are tracked in [19] based on the proximity of users and user-feedback about social networks in order to find common individuals that may want to be introduced. Both of these projects are examples that use user-profiles and Ubicomp techniques to aid individuals in some social function. In general these projects and other similar projects that use profiles exist but do not propose a general model or framework for building context-aware applications to aid people in social functions. Instead they focus on supporting social functions within the specific domain of a single application.

4. Related Cognitive Theories

The Computer Supported Collaborative Work community has investigated how context affects collaborative work for some time now by studying various models of human cognition. Some of the more notable models that have been investigated are Activity Theory [11], Distributed Cognition [8], and Situated Action [18]. These models are used primarily as a tool to help study how individuals and groups relate to the various contexts in the environment within which they work in hopes of designing new systems that support more fluid collaborative work.

4.1. Situated Action. Situated Action as its name implies is the study of human activity in situ. The theory focuses on uncovering the human learning process when unplanned improvisation is needed in the face of ever changing contexts. Situated Action research is conducted on the relatively short-term granularity of a situation. The important contribution of this theory is its stressing of the fact that although a great deal of activities carried out by humans is pre-planned there is still quite a bit of variance to plan when an individual is confronted with a specific situation. This is also one of it's shortcomings as it does not concern itself with the original intent based behind the plans which may have bearing on why the situation occurred to begin with.

4.2. Distributed Cognition. Distributed Cognition(DCog) also studies human cognition within the context which it occurs; however at the granularity of a system of people and artifacts. Hutchins uses an example as the navigation room of a ship as the example of one cognitive system, and studies the flow of information as it intermingles between humans and the various tools used in ship navigation. The primary focus then is on how the process of navigation incorporates new knowledge from people, and knowledge that is built into the tools being used, along with how information is transferred between tool and user. This provides the opportunity to reveal more about the information processing aspects of a system. However, it does not place specific emphasis on the conscious intent of the people in the system. Essentially, the theory treats people and artifacts as the same, inputs and outputs for data and does not place a large emphasis on the intent of people within the system.

4.3. Activity Theory. The study of activity is a branch of psychology that developed in Eastern Europe and Russia. It studies the activity of a person or group of people as enacted within a given context. The model of Activity Theory includes seven elements, each of which has affect on each other through mediating relationships. Figure 1 is a slight modification on a figure presented by Kari Kuutti in [11] that shows the seven elements and how they relate to each other: the subject (user), the object (objective) of the user, the tools present in the environment that can be used to aide in achieving the object, the rules that may be present within an environment (mores and norms), the community of other users that are near the subject, the work division or way that a cooperative task is broken down between the community, and the outcome, which is the ultimate result of the person working on the objective until they have reached their goal.

All of these entities exist within the context in which the activity is being performed, and hence they are mediated by the context, and contribute to the context at the same time. As depicted in figure 1 some of the elements in the model are connected by mediating relationships. A mediating relationship is when two things have an effect on one another such as when the subject wants to work on the objective. The tools that are present in the environment contribute to the context within which the activity takes place and mediates how the subject works on the object.

Finally how a user completes an activity can be analyzed by further breaking down how they achieve the tasks that in composition create the activity. Overall an activity is being completed to meet the needs of a specific motive. Motives are achieved by completing several goals that contribute to realizing the motive. Activity is the unit of work that corresponds to a motive, an Action then is the work required to carry out a goal. Actions are aggregates of different routinely carried out minute steps that do not require conscious effort; each step is known as an operation. Actions and operations are directly affected by the short-term situation in which they are being completed.

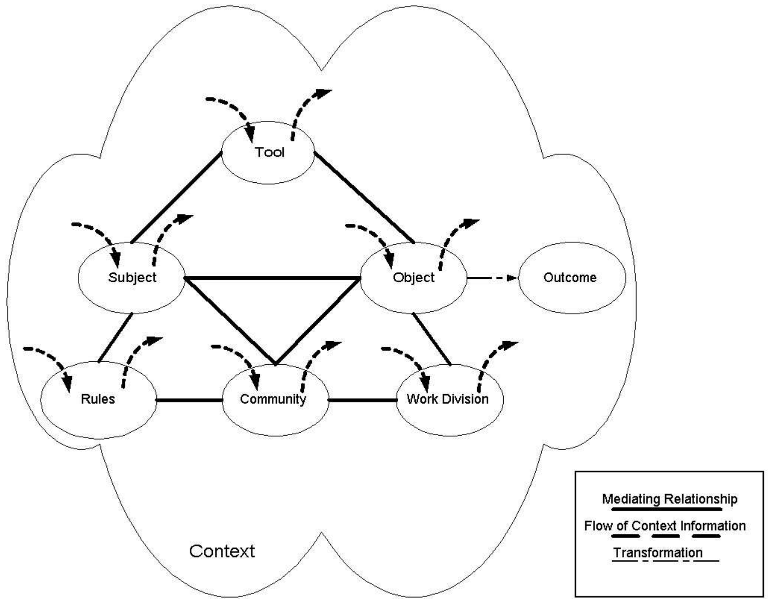

Figure 1. Overview showing the elements of Activity theory, and the mediating relationships between them.

4.4. Comparing the Cognitive Theories. The three cognitive theories in this section were all created with the same motivation, to understand human cognition. There exist significant differences when comparing the theories in terms of the following: The general scope of the unit of analysis in terms of time, and focus on human factors such as planning for objectives. The explicit structured support and categorization for elements that influence group cognition. How much emphasis is placed on analyzing what affects an individual.

Situated action focuses as it's name implies on the unit of analysis of a situation. Analyzing only a situation implies that cognitive actions taken are only studied within the context of the moment in which they occurred. An activity from Activity Theory lasts until the outcome is achieved; this makes provisions for the existence of long-term goals and contexts which cut across situation boundaries. Goals are a very human element (excluded from DCog, and Situated Action), and should be factored into system design. It is both spontaneity and planning that contribute to the realization of an objective, and hence both should be considered when designing systems that better suit human learning, and routines.

The primary unit of analysis for DCog is the cognitive system, it is unclear how a situation or objective factor into this.

Out of the three theories only Activity theory contains structured support for analyzing group cognition, and presents relationships between the basic entities that affect group cognition. As shown in Table 1 some of the entities from activity theory are considered for analysis although not as explicitly in DCog, and Situated Action. The structured support in Activity Theory for cognition under the influence of a group provided an organizational framework that was useful for helping in the creation of the model presented in this research. It is thought that it would also be easier to use in studies aimed at eliciting requirements, or design decisions for software geared towards supporting groups due to it's structure.

Activity Theory unlike distributed cognition and situated action places a great deal of emphasis on the actor [7]. Ubiquitous computing should be user-centric, and should not treat the user simply as another input/output in a system as per the distributed cognition theory. In order to fluidly support users while minimizing distraction to them the technologies involved must support the users goals in a manner that does not complicate or obfuscate their completion. Furthermore some human understood contexts such as roles can at times affect how a situation influences the completion of an action, and this should be considered in analysis.

	Tools	Rules	Subject	Community	Work Division	Objective
Situated Action	No	No	Yes	No	No	No
Distributed Cognition	Yes	No	No	No	No	No
Activity Theory	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. Support for entities in Situated Action, DCog, and Activity Theory

4.5. The Use of Activity Theory in the Design of System Models for Supporting Groups. Activity theory was used to develop the conceptual group model in this research primarily for the fact that it's entities enumerate the basic general constituents that affect how groups interact to complete tasks. The various entities in the theory provided inspiration for the investigation of a new set of general contexts that the group model in this research was designed to investigate the use of. Based on the inclusion of a Tool entity we developed the concept of a device group as a tool for supporting group interactions. In part the organization of the entities was used to determine how to delegate responsibility for handling of the entities between the system layer, and application layer of our model.

5. Usability Testing

Two types of usability testing were conducting to complete this research. Heuristics evaluation performed by the researcher, and scenario based testing with additional end-users. General guidelines for both types of tests were found either in Jakob Nielsen's Usability Engineering book or from sources online. Approval was obtained from the IRB on ASU campus based on the test materials and overview given in appendix refappend:materials.

5.1. Heuristics Evaluation. A heuristics evaluation involves reviewing the user interface for your software and evaluating them against the following general usability guidelines:

1. Simple and Natural Dialogue - Interfaces should be designed to contain just enough features for users to accomplish their tasks. Furthermore those features should fit in a context that matches the task at hand to be easily understood.
2. Speak the User's Language - Terms in interfaces should be written to be easily understood based on the domain of the target user.

3. Minimize User Memory Load - Do not require the user to remember too much in order to accomplish a task. If a large amount of information is needed to complete a task shortcuts and features should make that information readily available.

4. Consistency - Application interfaces should be feature consistent with other applications in the same domain or software suite.

5. Feedback - Enough information about the current system state should be given to the user in order for them to feel in control.

6. Clearly Marked Exits - Quitting an application should be simple.

7. Shortcuts - Provide streamlined ways for experienced users to complete highly used actions more quickly than common users.

8. Good Error Messages - Give sufficient information for user's to easily diagnose and resolve usage errors.

9. Prevent Errors - Interfaces should be designed so that situations where usage errors occur are minimized.

10. Help and Documentation - Systems with enough features, or sufficiently conplex interfaces should have a well documented manual or online help system.

A heuristics evaluation should be performed before a usability test involving users to resolve any issues the designers or developers may have overlooked that would interfere with further testing.

5.2. Scenario Testing. Scenario based testing involves having actual end-users use the system to perform a specific task or set of tasks to complete a goal. Scenarios are generally given on paper and set the context for a task. Tasks are generally completed in a lab with a test facilitator. Some general guidelines for carrying out scenario testing that were followed for this research are proposed in [3] and [13] :

- Testing should start with a round of one-on-one scenarios between a facilitator and a single user.
- Around 30 minutes per a test is considered optimal to uncover most shortcomings of the system, and make the best use of a user's attention span.
- A small group of 4-5 users is all that is needed to carry out testing in order to uncover 80 percent of the usability defects.

Scenario testing is conducted with users and a test facilitator. The primary role of the test facilitator is to perform the following tasks prior, during, and after testing:

- Provide background on the project to test users.
- Discuss with test users the processes which will be used for testing.
- Walk the test subjects at a high-level through the testing tasks to be performed.
- Answer questions about the test tasks as the user completes them.
- Instruct on how test users are to give feedback during testing.
- Monitor users as they use the test the software.
- Although it should be somewhat avoided the test facilitator should provide the test user with some hints on how to use the system in the event that the user begins to becomes frustrated.
- Perform post-test debriefings.
- Thank the test users for their participation!

6. Summary

The concepts of research in the area of Ubiquitous Computing in general and how it pertains to this reasearch were re-iterated here. Research for supporting groups in a Ubiquitous Computing environment was also reviewed. Theories of human cognition that created inspiration for portions of this research were then presented. Finally usability testing concepts, and guidelines that were followed in order to conduct testing of prototype software for this research was defined. The next chapter presents a conceptual Social Group Model for supporting groups of users in a Ubiquitous Computing environment that forms the core of this thesis.

CHAPTER 3

The Social Group Model

This chapter presents a conceptual model for the purpose of supporting the development of context-aware applications that support groups of users as they collaborate [21]. First some background and motivation is given, along with definitions for the concepts of a Social Group, and Social Context. Second the challenges of supporting groups in a Ubicomp Environment are described. Next the System Model which encapsulates the underlying assumptions used to develop our group model is presented. The group model which describes the support of user groups through the use of device groups as a tool is presented next. Finally how Activity Theory provided inspiration for the creation of the group model is given in conclusion.

1. Supporting Social Groups in a Ubicomp Environment

Supporting the tasks for groups of users acting in unison will be a pivotal part of allowing technologies to seamlessly fade into the background in a Ubicomp environment. Interacting with other people is a fundamental part of work and play. Although supporting group interactions is intrinsic to pervasive computing principles, problems specific to supporting this using context-aware computing is an area of growing interest to researchers.

In order to develop a system model to support group interactions the definition of a group as supported by the model should first be defined. From [12] a group is "A number of individuals, defined by formal or informal criteria of membership who share a feeling of unity or are bound together in relatively stable patterns of interaction". Groups of users based on this definition are referred to as social groups. Two key factors are elucidated by this definition. First, groups of people share some sense of membership. Second social groups exhibit routines that can be described with context, and can be predicted based on those contexts.

1.1. Social Group Membership. Individuals within a group will share a feeling of membership that can either be informal as in when meeting with friends at a party, or more formal such as working on a software development team. In many cases the individuals view of their membership in the group, and the understanding that they are a member by the entire group as a whole will be in consensus. Establishing the rules for membership in a non-formal group can be a non-trivial exercise. For many formal groups where membership is assigned computerized support is more easily created.

The time an individual spends as a member in a group is termed the lifetime for that persons group membership. Also, the time that a group is active based on the presense of at least one group member will be referred to as the lifetime of the group. It should be noted that both the lifetime of a person's group membership, and the lifetime of a group in general will cut across computer supported sessions.

In Activity Theory the Subject, Rules, and Community have mediating relationships with eachother. This indicates that social group memberships are also a source of context that can contribute to mediating a situation with a social group. Generally there are social norms and mores that define how a group will or should react to a specific person's role or skillset. For example when a person is in the group of managers at a business, this fact contributes to how group members interact with them.

1.2. Predictable Interactions. A community in Activity Theory relates to a group objective through work division. Work division provides a useful means for predicting group interactions based on the fact that groups generally organize themselves so that interactions occur at a designated location, then start and end at scheduled times. Team meetings or classes are examples of group interactions that are predictable. Time, and location are not the only contexts that can be used to predict an interaction. For example, a software engineer may inform his manager that a report is complete and available before it is due as he passes him in the hallway. This communication based interaction occurred based on the fact that the two shared membership in a group working on a shared objective, the report was complete, and the identity of the person in the hall based on role and work division required the report next.

The beginning or ending of a group interaction to work on an activity will not always be denoted by contexts that can be measured using sensors. A means must be constructed to characterize these more social based situations and denote them by organizing and storing other contextual information. This information will be a mixture of detectable contexts from the environment and some specified by the end-user. Although we desire to automate system support of group interactions, attempting to provide full automation may not be desirable in every instance. For especially sensitive or complicated interactions the correct amount of feedback must be given to, and extracted from the user in order to properly minimize distraction. Throughout this research the use of social contexts that can be used by applications to predict group interactions is discussed.

1.3. Social Context. A social situation requires more contextual attributes than those that are provided by sensor measurable physical phenomenon such as noise, light or location. Arguments such as the one in [17] state that researchers are not leveraging every available source of measurable information that contributes to the context of a situation. The model presented by this research was designed to include context attributes that are detectable through sensors, and user defined. Contexts used to describe group dynamics and interactions are referred to as Social Contexts:

Definition 1 Social Context: *Information relevant to the characterization of a situation that influences the interactions of one user with one or more other users.*

It can be seen that more fully explored contexts such as time and location affect several entities in the Activity Theory framework. The location that a group is at may limit the availability of tools, alter Work Division based on available Subjects, and change the scope of an objective. An example of location used as a social context involving chat is given in example 1. The affects of time on a group's schedule may also lead to similar consequences when analyzing entities. Social contexts also have an impact on the entities that are used to study group activity.

Example 1 *Consider an application built for the activity of online chat that uses location as a social context. A user may specify that they only want their laptop to join a chat group with family and friends when the context indicates a socially acceptable place to chat, such as their home. Hence, when in the office the chat application would not automatically log the user in. But, once the user is home the chat application would start and allow the user to talk with friends.*

Another context that changes how entities interact with each other is the skills of a Subject. Qualifications to use a specific tool may dictate what tools can be used to achieve an object. Similarly the skills of the subject may fit a specific need in the work division which, would need to be adjusted otherwise. The skill-set of a Subject then contains examples of social contexts which mediate the entities and the relationships between them as well. This research wishes to create a model that has a means to capture these new contexts in hopes of fostering interest in having researchers catalog and study using such contexts to improve computer usability, and create new applications.

2. Challenges of Supporting Social Groups in a Ubicomp Environment

There are numerous challenges inherent to supporting Social Groups. The following example will be used to illustrate the specific challenges addressed by this research:

Example 2 *Consider a group of students taking a software engineering course that are involved in regular meetings to peer review reports. Assume that each user carries a PDA that runs software allowing the users to participate in some computer mediated collaborative review process. When members are in attendance at the designated location, at the correct time the application should start the peer review process. At some point during the review if the author and a reviewer choose to leave to ask their instructor a question they should be able to continue working with one-another independent of the other group members. Similarly when they return to the peer review they should be integrated into the group as a whole again seamlessly. Then once the review process is over the software should end the review process and ensure that everyone in the review has the final version of the document.*

The challenges our system model is built to address from example 2 are:

1. **Determining when a group interaction will occur -** Within the membership lifetime of the group several collaborative interactions will take place. Context-aware software tools must detect the start of interactions to allow computing devices the ability to form communication links between each other, and support users with context relevant functionality anytime/anyplace.

2. **Determining when an interaction is going to end -** Detecting the end of a group innteraction to support any actions needed when the interaction ends is another fundamental challenge similar to detecting the beginning of an interaction.

3. **Supporting interactions despite varying network-infrastructure capabilities -** For a truly ubiquitous software solution to support groups in a Ubicomp environment software should support collaboration between devices in infrastructures with varying network capabilities. This makes it necessary to design a solution that implements the group model which allows devices to function in an ad-hoc network.

In the activity theory framework we see that how the subject works on the object is mediated by tools, rules and the community. How the community works on the object is mediated by the subject, rules, and work division. The goal then is to develop a tool that supports the following:

1. The mediating relationships that affect how the subject works on the object.

2. How the community is mediated while working on the object.

3. Addresses the above challenges given in this section for basic group properties.

The System Model that is used as the basis for developing such a tool is given next.

3. System Model

The Group model was designed around the following assumptions:

1. Each user will carry one or more Ubicomp devices that can communicate over a wireless network.

2. Per challenge 3 and the previous assumption, users will collaborate using peer-to-peer collaborative software.

3. No byzantine failures will exist within the system, implying that interactions will happen between trusted parties.

4. Clocks between distributed Ubicomp devices will not drift outside a specific tolerance level, with tolerance being application specific

An effective model for a system that supports the activities of social groups in terms of managing the user's perception of membership in groups and group interactions must minimize distraction [4]. Reducing distraction is not only accomplished through automation, and the model was not designed to only include systems that provide full automation of resolving group membership, and group discovery. Decisions involving membership must be reached through consensus with other group members, and establishing group membership is non-trivial and should require user intervention. Some groups may wish to have their discovery kept secret and would not desire for an outsider to join their group through automation. In general the social factors for group membership are complex, and user feedback should be obtained at some point in the process.

General routines for device setup, and group membership after initial configuration should be automated in order for a system to maintain more time out of the foreground of user focus. The model should also be constructed so that the roles of users can be used as in traditional system design to limit the functional scope of an application. Now a model is presented to support group activity that adheres to the criteria mentioned up to this point.

4. Social Group Model

This section presents a conceptual model based on the challenges in section 2 that can be used as the basis for developing systems and frameworks to support group activity. A goal is to require minimum time expended on system set-up for supporting group interactions when working towards the objective of an activity. The model is focused on supporting two types of groups; First, the user and their social groups (subject and community). Second, the devices that network together as a tool for supporting the actions and operations group members take to work on the objective.

User Group: The User Group is a social group that the end-user interacts with to complete the objective of an activity.

A user-group will exhibit stable patterns of interaction to work on activities. The time taken to complete a routine interaction is referred to in this research as a User Group Session.

Definition 2 User Group Session: *The time period during which users are interacting with one another to perform actions, or operations as part of an activity.*

The team meeting in example 2 demonstrates a user-group session. Team meetings generally occur at a regular and reoccurring location and time. A more spontaneous user-group session would be friends collaborating on a task from different ends of a bus while traveling.

The second group that is focused on in the model is formed by devices carried by the user and their group-members called a Device Group.

Device Group: A device group consists of several devices used by a user, or user group that network together and act in unison to form a tool for a group to use when working on completing actions or operations as part of an activity. A user group can have one or more device groups active to support their activity. There must be a means of determining which devices belong to what device groups. Also it is necessary to determine which device groups to activate in order to support the actions and operations needed by user groups. In order to aid in determing which devices belong to the device group used to support a user group's action the model uses the context of the device.

Definition 3 Device Context: *Any detectable and relevant attribute of a device, its interaction with external devices, and/or its surrounding environment at an instant of time.*

A device context includes contexts that are either detectable through sensors, or user-defined social contexts stored on the device. In order to use the device group as a tool that seamlessly supports user-group sessions during an activity social context data from the subject, community, tools, social rules, and work-division can be configured by the user or user-group and stored on the device. By allowing the device group to support group actions users will feel that the device group as a tool understands the needs of their user groups.

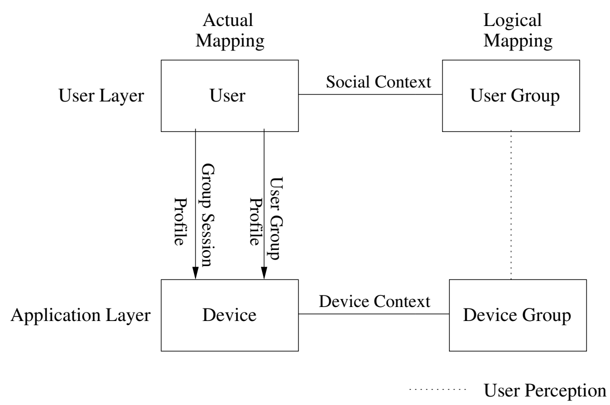

Figure 2. Multi-layer group model in Ubicomp environment.

Figure 2 depicts the relationship between the User, their devices, the User Group, and the Device Group. The group model is depicted in two layers: the User Layer, and the Application Layer. The user layer includes the user and their groups and shows that a user is a member in a group based on Social Context. The application layer represents the software running on users various devices, and shows that based on device context a device is a member of a device group. The dotted line reaching from the User Group to the Device Group indicates that the group of users will perceive that the device group was formed based on the social context and is aiding them.

The final mapping shown in Figure 2 between the User Layer and Application Layer is from the User to a Device. This mapping is done using profiles, a profile is nothing more than a data file that resides on a device that is used to specify contextual data. Profiles are used to store social contexts, and are used by a device to customize application functionality and form device groups. Profiles have been used to support social functions in [5] and [19], and the techniques presented by those projects could be used as a means to mine or refine the accuracy of context data that is placed in profiles. There are two profiles in the model: a User Group Profile, and the Group Session Profile.

User Group Profile (UGP): This profile contains social contexts about the user that are relevant to a user group. The User Group Profile includes a unique user identifier, user preferences, and any additional social contexts used by applications such as the role, skills, schedule or interests of the user within the group. This data is then used by applications to customize functionality for the user in order to tailor specific actions while interacting with a group. A software engineering team member's UGP from example 2 may contain the following contexts:

1. Username - An alias used to identify the user when using collaborative software.
2. First Name - The user's first name.
3. Last Name - The user's last name.
4. Email Address - A current email address.
5. Team Role - The designated role within the software engineering team.
6. Interests - The user's hobbies.

A UGP once configured by the end-user could be: ("Johnb123", "John", "Bodily", "John.Bodily@someisp.com", "Team Lead", "Motorbikes")

Group Session Profile (GSP): Contexts that indicate the beginning and ending of group sessions are placed within a group session profile. This will at least but not be limited to include a timespan indicating the length of a user-group session, and the location where the session will occur. "Anytime", and "Anywhere" would be valid values to support sessions that do not have a specific time, or location. The Group Session Profile is used to control which devices owned by the community take part in the device group to support the user group. From example 2 a GSP to describe the team's document peer-review session could contain the following:

1. Group Identifier - A unique identifier allowing system software to disambiguate this GSP from any other in the system.
2. Group Name - A canonical name assigned by the end-user to identity a group interaction.
3. Group Purpose - The purpose of the group session.
4. Location - The physical space within which the group interaction will occur.
5. Start Time - The time when the group interaction will begin.
6. End Time - The time when the group interaction will be complete.

A GSP for the team meeting that follows from the above list would contain: (255, "SoftEngTeam", "Document Peer Review", "Computer Commons", "4/16/2006:01:30:00", "4/16/2006:03:00:00").

5. The Group Model and Activity Theory

Activity Theory proved useful when developing the concepts behind the group model as it provided a well defined and structured framework to study for background about what social contexts influence group activity based on the mediating relationships between the subject, community and other entities. This section will describe how the subject and community entities in Activity Theory relate to the group model in terms of what contexts will reside in profiles.

5.1. The Subject, Community and Profiles. **Subject -** The UGP is a means to encapsulate the contexts that are important to the subject. As mentioned this will include information used to identify them to applications and nearby users. It can also contain data that is important to mediating the relationship between the subject and other entities. For example roles that influence social norms and rules held by the community, and qualifications for operating specific tools. Interests or even the purpose of the activity that the user has in common with the community can be placed here as well.

Context data specified in the UGP for the subject is seen as being used to mediate work on the object using applications in two ways. Some social contexts will be used to influence what functionality may be provided to the user. From example 2 a user with the moderator role may have access to a feature that allows him to control rules for who in the community has access to comment or edit the shared document under review. The second way UGP information could be used is in tandem with the GSP as a means to aid in determining when some group sessions with the community will begin or end.

Community - Using the contexts in a GSP is seen as a means to support the community by triggering the formation of a device group to support and mediate them as they interact with each other to work on an object. As mentioned previously it is sufficient for many interactions to be characterized simply through use of location and times per the schedule for a communities work division on an object. However, there are means to use contexts important to the community along with those of a subject in the UGP to characterize the start or end of an interaction.

Example 3 *For example a session may be launched based on a UGP being found that possesses a needed skill or role that the group is seeking. Consider a physics study group looking for someone nearby with the role of "physics tutor" in order to hopefully start a chat and get a question answered. In this case the study group is collocated and may not need to chat over* *a computer. All of the students in the group will want to be able to ask questions remotely once a tutor whom is not colocated is found as part of working on the activity of study.*

How the community mediates the subject, and the work division mediates the subject as a member of the community to work on the object will be application specific. In general Activity Theory when used with an ethnomethodelogical study can be used before work on an application begins to determine how users in a community cognitively expect an object to be transformed into an outcome. Study results can then be used to determine what social contexts are important to the subject and community when trying to complete an objective. The discovered contexts can then be placed into profiles, and used to formulate requirements, and test plans for how contexts will be used to automate functionality, and form device groups.

6. How The Device Group Supports User Perception

A system designed on the social group model will support the user's perception that the device group supports their user-groups in two ways. First, using the GSP to automate device group formation to support user group routines as they work on an activity. Second, using the UGP to support the user's social contexts and preferences in order to provide functionality specific to their needs based on the activity, or actions needed to be completed at hand.

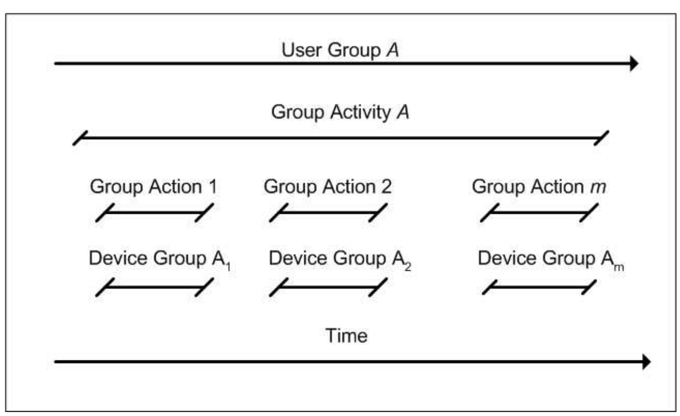

Figure 3. Device Groups form as the User Group meets and parts to work on an activity.

Figure 3 shows a graphical representation of times that the groups from the group model are being formed, and dispersed as viewed by a group member over time. If a user is a member of the User Group A and is working on the Group Activity A during the time as shown in Figure 3, they will participate in m routinely held user group sessions as they perform m actions on the object of the activity. Each device group A_i is shown in the figure as being active from the time it is formed, to the time it is dispersed based on the actions taken by the group. Thus the context representation from the Group Session Profile after initial creation will be used m times to form a device group for each of the m user group sessions.

A device group is a collaborative tool for supporting groups as the work on the objective of an activity. A user is a member of a user group based on social contexts, which also determine the activities the group will be collaborating on. Device groups use a device context to join the device groups that support the user group. The device context includes social contexts by having applications use a UGP, and GSP. The influence of Activity Theory on the group model and applications that could be developed to use device groups is presented in the next section.

7. Conclusion

In this chapter the properties of a social group from the standpoint of sociology was reviewed first. It was shown that in order to support them using computing a model that supports membership and routine interactions should be developed. Next the concept of a social context was defined. A social context defines information relevant to the situation as experienced by the group and it was seen that it too could be used to automate and provide useful functionality for users. The challenges of supporting groups in terms of automating the beginning and ending of routine interactions, and supporting them andywhere/anytime were reviewed.

Next the Social Group Model that forms the core of this research was presented. The model presents a concept to support user groups based on the use of profiles to encapsulate social contexts and feed them to computing devices. The social contexts then add to the context used by devices to network together in order to support the group as they perform actions and operations on an activity. How Activity Theory was used as a basis and inspiration for defining the profiles to capture and use contexts for the subject and community was presented. Finally a description of how the networked device group creates the perception that it is a tool that understands social context by using contexts important to group purpose to automate device networking was reviewed in greater depth.

Prototype system software designed around the social group model has been developed to test the feasibility of using the model to build software. A sample application that allows user groups to chat based on groups formed using the UGP, and GSP was created. The next chapter gives an overview of the prototype that was then used to explore through usability testing how actual users expected software built on the model to function.

CHAPTER 4

A Prototype Group collaboration System

In this chapter an overview is given of the prototype system software that was developed for this research based on the conceptual group model from the previous chapter. Per the Group Model this software is a testbed for the support of experimenting with the creation of peer-to-peer group collaboration applications. The developed prototype includes the following: The Situation Aware Ephemeral Group(SAEG) management service that uses the GSP to form device groups. A service for User Group Profile Distribution(UPD) between PDAs. An application named the Group Manager(GM) which is used to create and distribute GSPs. Finally to experiment with group collaboration a Group Chat application to support users text messaging and sharing files was created.

The prototype uses a User Group Profile that contains the following relevant information about a user:

1. **Username -** A username that the end-user wishes to use as a nickname.

2. **First Name -** The first name specified by the end-user.

3. **Last Name -** The last name as specified by the end-user.

4. **Email Address -** The email address used by the end-user.

The Group Session Profile that is created by the GM for use by the prototype contains the following information to determine when to start or stop a group:

1. **Group Name -** A name assigned to a user group by the end-user that crates the GSP.
2. **Location -** An identifier that indicates the location for which the user-group session will become active.
3. **Begin Time -** The start time for a user-group session.
4. **End Time -** The end time for a user-group session.

1. Platform

The details about the platform used for software development and deployment are recorded here. All software was designed and built to deploy on the Dell Axim x5 PDA running Microsoft Pocket PC Version 4.2. The specific hardware specifications of the PDAs used are listed below:

1. **Processor Type:** Intel XScale
2. **Processor Speed:** 400MHZ
3. **RAM Size:** 64 MB
4. **ROM Size:** 48 MB
5. **ROM Version:** A05
6. **Wireless Card:** Dell TrueMobile 1180

All software development was performed using the Embedded Visual C++ 4.0 IDE.

2. Implementation Details

More detailed information is presented here about the individual applications and services of the prototype. A brief overview about the functionality provided by the SAEG and UPD is given first. Next a more detailed description of both the Group Manager and Group Chat is given. The Group Manager and Group Chat demonstrate the use of the UGP and GSP profiles from the model, and detail the basic architecture of applications developed for our prototype.

2.1. Situation Aware Ephemeral Group(SAEG) Service. The SAEG is a service for building device groups using an ephemeral group-view maintenance service that organizes devices by their I.P. address in an ad-hoc network into a group view that can then be used to send application message over a network based on a group identifier. It implements a non-deterministic group-view maintenance protocol for building a list of device addresses for a group. Group views are built based on the contents of the Group Session Profiles on a device. Each device with the same profile that is in range of another device group member will be a member of the group.

2.2. User Profile Distribution(UPD). The UPD service is used to distribute a User-Group profile between devices. It's protocol is designed to detect new neighbors and exchange the User-Group profile with that neighbor. Time-stamping is used to nvalidate UGPs in order to aid in detecting when a user is still in range. When a user goes out of range the User-Group profile is removed.

2.3. Group Manager. The Group Manager(GM) is an authoring tool for creating and distributing group sessions. A group session is created and then represented by a GSP. The user can then distribute the GSP to nearby users that are identified by their UGPs. The GM also manages the persistent storage of groups. Deactivating sessions with a group if no interaction is desired with that group despite the context for activating the group being met is also allowed. The design of the application architecture is show in figure 4 and described now.

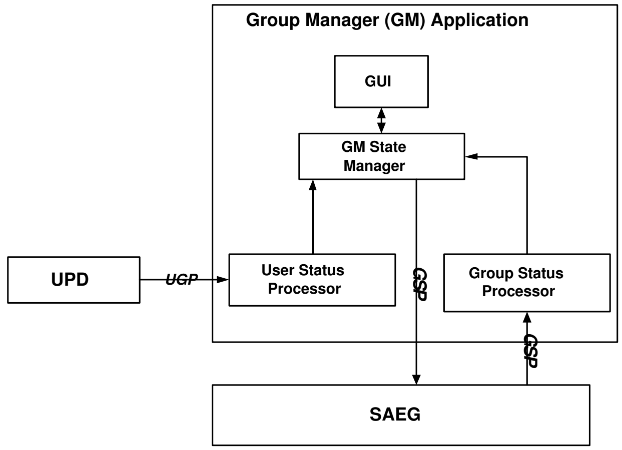

Figure 4. Architecture of the Group Manager application.

GM State Manager: All application state is controlled by the state manager. This includes inserting, editing, and removing GSPs from persistent storage. The state manager also maintains a list of nearby users that it has received a UGP for. The current operative status of groups that the device has obtained a GSP of is also tracked.

Group status is managed based on input from the SAEG, and from the end-user. A group can have one of there operative statuses:

1. **Stopped -** The end-user has told the SAEG that they do not currently want to be a member of a device group no matter what the current context is.

2. **Started -** The SAEG is given a GSP in order to place a group in a state where the device will be a group member when the context in the GSP of the group is met.

3. **Active -** Indicates that the group session has been started, the context in the GSP has been met and that the SAEG has made the device a member in that group.

User Status Processor: Notifications are received by the GM about active users in the vicinity by the User Status Processor. When a new profile is detected based on input from the UDP the correct operations on the GM State Manager will be invoked to allow the application to notify the end-user of that user's presence in the vicinity. Similarly, when a

UGP has not been received after a specified time an operation on the GM State Manager will be called to remove the user corresponding to the UGP from the application.

Group Status Processor: Operative status information from the SAEG is used by this component to set the state for a group using the GM State Manager. The list of active groups is received from the SAEG, and used to set the active status of a group. Groups only appear in the list from the SAEG when they are active. Hence when a group no longer appears in the list of groups from the SAEG it will be marked as started.

GUI: The interface for using the GM is shown in Figure 5. The primary interface displays to the user a list of groups at the top that are currently stored in persistent storage or in memory. For each group the GSP is displayed, along with an icon much like a light from a stoplight that indicates status. Red is stopped, yellow started, and green active. Also the list of users in the nearby area is displayed in the primary interface allowing the user to determine whom it is available. The user can view their current location by selecting a menu option as well.

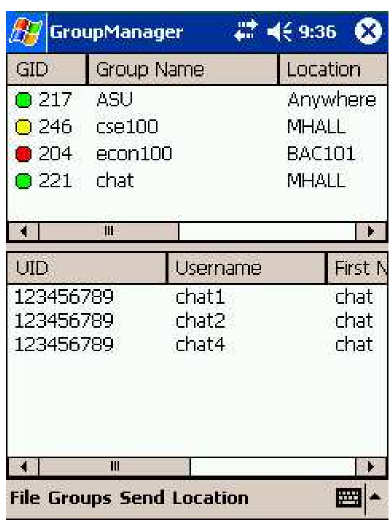

Figure 5. Screen-shot of the prototype Group Manager application.

2.4. Creating a Group Session Using the GM. Creating group sessions is done using the GM. When creating a group session the user will first be asked to configure a group name, and a location used to activate the group. By default the current location is selected and displayed. The user can choose to change the location to a new one which they know the identifier for, or select an "Anywhere" location.

The next step involves configuring a start and end time for the group interaction represented by the GSP. When the user first views the interface the current time is used to specify the start time, and the end time is configured to be an hour later. The user can then set the start and end times to indicate the situation they wish, or specify an "Anytime" context. Upon completion on specifying the time the GSP is sent to the SAEG and used to place the group in the started, or active state.

2.5. Distributing and Receiving a Group. In order to distribute a profile after creation the user must select the profile and then select the "Send Group" option which is under the "Send" menu. Next the end-user is allowed to select one or many nearby users in order to send a GSP to. A GSP for a group is then sent over the network to the GM that is running on the remote-user's PDA.

Receiving a group is very straight-forward. when another user desires to send a group across the network to you a message will appear asking if you would like to review the profile. If the end-user possibly wishes to join the group they can review the GSP for the group and accept or cancel adding their device to the group. When accepting a GSP the device is not automatically put into the group, instead the user must start the group if they really want to join it.

2.6. Group Chat. The group chat application as it's application name implies allows groups of users the ability to talk and exchange files in ad-hoc networks based on group sessions defined in GSPs. The chat application retrieves information needed to send messages to groups from the SAEG. Information about users is obtained via their user-group profiles from the UPD.

Users are organized into social groups as defined by the GSPs on the device that are registered to the SAEG. A group in this case is analogous to a chat channel. A user is included in a specific social group if they have a valid user-group profile and a shared group-session profile with other users. Users can send private messages, broadcast messages to the social group, and share files. A user is also allowed to review the user-group profile of other users, and the group-session profiles of the various groups they are members of. Additionally after performing a heuristics evaluation a function that allows the user to review the current location was added. This was done to provide feedback to the user about the context that has either activated or deactivated a specific social group. The basic components of the chat application are shown in Figure 6:

Group Chat State Manager: The Group Chat State manager updates the application data needed to characterize the state of activated groups. It uses the UGP to determine which interactions the end-user should be allowed to perform on a per group basis, and GSP to determine the user group sessions. All text sent to, or received from a group is buffered here for retrieval by other GroupChat components.

GroupUpdate Processor: This component is responsible for registering the application with the SAEG service. It also processes notifications sent from the SAEG about changes to the the current set of active user groups, and to changes in the membership views for those groups. The results of processed notifications are used to update the ChatGroup Manager's state.

ChatMessage Processor: The ChatMessage processor interacts with the ChatGroup manager and the SAEG service to send, and receive messages, or files destined for a remotehost in a group. Messages still needing to be sent are extracted from the ChatGroup Manager and passed to the SAEG service, and vice-versa.

GUI: The application GUI provides the user with a display of the currently available groups that are included in their set of currently active group sessions. The current membership views for those groups is displayed in a tree beneath the group. Text messages are shown on the right of the screen, with the text from only one group being shown based on which group the user has higlighted on the left. Again for consistency the user is given an option that allows an end-user to see their current location.

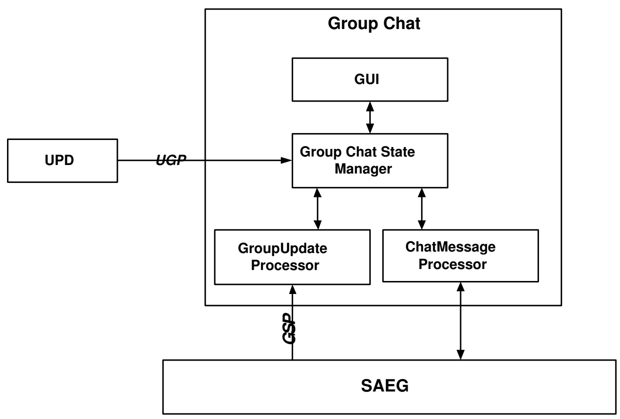

Figure 6. Architecture of the GroupChat application.

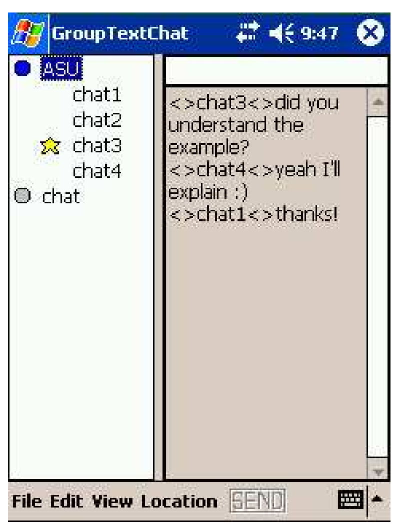

Figure 7. Screen-shot of the prototype GroupChat application.

3. Summary

A prototype system implementation for supporting group collaboration based on the social group model was reviewed in this chapter. A Situation-Aware Ephemeral Group membership management service for forming group views based on aggregating I.P. addresses using GSPs under a group identifier was overviewed. Group identifiers are used to send network messages to all devices in that group with an application registered to receive messages from that group. UGPs are distributed using a User Profile Distribution service that exchanges profiles with devices owned by nearby users for use in applications.

Next the functionality, and architecture of a Group Manager application used to define, distribute, and store GSPs was covered. GSPs created by the GM application currently consist of a group name, location, and a start and end time which is used to indicate the start of a group session. A GSP is then distributed to other users nearby for use with an application. GSPs can also be edited, deleted, and turned off. The GM also displays to the end-user whom is nearby, what groups are stopped, started, or activate, and what the current location of the device is.

Finally a chat application named the Group Chat was presented as a sample application built upon the prototype system. The application supports exchanging text messages between a single user and a group or between single users. File transfer between group members is also allowed. Groups are defined based on GSPs, and users are identified based on their UGPs.

The primary purpose of the Group Manager, and Group Chat applications was to present the concepts, and sample architectures of applications that use the social group model. Usability testing was conducted on the Group Manager, and Group Chat in order to evaluate how users feel while using software built on the model. Feedback from the tests is discussed in the next chapter, and includes: General problems found during testing. Suggestions about how to improve our system to better leverage the group model, and additional ideas for applications that the model may be used to build.

CHAPTER 5

Usability Evaluation

1. Introduction

Usability testing of the prototype described in chapter 4 was performed in order to explore the usage of a system built around the social group model. Testing was motivated by the need to explore and determine what usability issues may exist when designing system software or applications to work with social groups and social contexts. The task scripts and questionnaires found in the appendix were used as the instruments to set background for the tasks, and collect user input. Input from users was examined and used to determine new uses and concepts that can be used by developing software using the model.

First, background about test-users, and test tasks will be presented. The collected data used to find that users took an average of 10 minutes and 36 seconds to create and join a group is shown. The usability problems associated with users not having proper feedback about contexts being used is discussed. Users wanted functionality to locate and form groups from two perspectives: group-centric and user-centric. The existence of user group sessions in an area was requested as a group-centric context to be used in order to help in locating and joining groups. Then, working with groups in a user-centric manner using the context of a user's formal group memberships and interests as a means to create informal group memberships is discussed. Finally, solutions as determined through synthesis of test results is presented and discussed as future work for continuation of research and prototype development.

2. Test Background

Testing that was conducted is overviewed in this section. Background about test users is given, and then an overview of the three test rounds that were conducted is presented. Tests included a heuristics evaluation, and two rounds of scenario based testing. Four users took part in scenario testing that was conducted on the 5th floor of the Brickyards in the Impact research group lab here at ASU.

2.1. Test User Profile. Four test users [13] were recruited in total to help perform usability tests. Due to the virtue that tests involved testing a research prototype it was desired for users to have some computer science background. The list below contains items about the basic desired background of prospective users:

1. Familiarity with chat programs would be beneficial in order to bypass or reduce the time needed to train the users in the basic mechanics of using a chat program.
2. Someone either already familiar with or more easily familiarized with the basics of context-aware computing. As this is a research prototype some domain-knowledge is desired in order to provide possibly more meaningful feedback about the design of our context aware group model, and software.
3. Understanding of computer user interface design would be likely to provide meaningful suggestions about how to improve the basics of our softwares user-interface.
4. Age and gender are not important for the purpose of this study.

Three of four users met all of the criteria from the above list. The fourth user was only familiar with chat applications and was included in hopes of obtaining somewhat of a perspective from a user that is not trained in computer science. All users stated that they used chat applications and various instant messengers on a routine daily basis. Additionally users mentioned collaborated with each other to play video-games, also three mentioned the use of blackboard to receive notifications about courses on ASU campus.

2.2. Test Rounds. The first round of testing was a heuristics evaluation. This was done to prepare the prototype software for scenario testing with test-users. Obvious problems with usability and consistency with other chat applications was corrected here. Additionally some enhancements that aided in usability were planned and made based on the results of this test round.

Two rounds of scenario testing were then conducted with improvements to the applications factored in from round one. Scenario testing started with one round of one-on-one testing between the user and the facilitator. Background about testing, and the project were given to the test user at this stage. The user was then asked to complete two small sub-tasks to collect data and familiarize themselves with the applications. A final round of testing involving all four test users using the system and applications together was conducted. The initial sub-task was similar to that conducted in the one-on-one round of testing, and then an additional sub-task to collect feedback on how users feel about group sessions reacting to changes in context was performed. The list that follows describes in greater detail the test tasks that were completed as part of scenario testing the software prototype:

1. **One-on-One:** A task performed between a test user and the test facilitator. The main goal of the task was to have the user create groups using the group manager. First the user is asked to create a group for the current time and location, and distribute the group so that they could chat with the facilitator. The second sub-task involved having the user create a group for use at another time, and location.

2. **Chatting as a Group:** Four users are asked to create a group and chat together all at once. This is similar to the first sub-task in the first task but tested with four users instead of only two.

3. **Leaving Groups:** The four users used the chat application as groups end and start due to changes in context from time and location.

Results of quantitative and qualitative measurements taken from the test rounds are presented in the next section.

3. Results

The focus of this section is to present the general findings after test completetion as it applies to software created around the model in Chapter 3, and supporting groups in a Ubicomp environment. Test results detailed in this section include quantitative and qualitative data collected from both a heuristics evaluation and scenario testing with actual users. A measurement of the time it took for users to learn how to use the group manager to create and distribute groups is first reviewed. Next the paramount importance of proper application and system feedback to the user for supporting groups and users using context-awareness is discussed. Observation and user feedback indicated that user's found forming a group difficult and wanted solutions that were both group and user centric in order to improve usability. In conclusion, suggestions for improving usability with social groups using our system and how it focused on the importance of using social contexts about users and groups to enhance proactivity is reviewed.

3.1. Time Taken to Learn Group Configuration and Distribution. Timing was collected in order to assess how easily users learned to utilize groups in the prototype as patterned on the social group model in table 2. It is felt that the time taken to learn the

User	Time (Minutes:Seconds)
1	13:24
2	9:59
3	9:26
4	6:55
Average	10:36

Table 2. Time taken to create, distribute and join a group

system was reasonable with an average of 10 minutes and 36 seconds. Previous to working with the group manager and chat application users were given an introduction on basic Power PC mechanics, and a brief high-level description of what functionality the group manager and group chat possessed. Collected timing data specifically included the time it took a user to configure and distribute a UGP, and GSP for the purpose of collaborating in a chat with another user.

Observation of actual system usage and feedback from users indicated that the most difficult and time consuming step was GSP distribution. During testing most users were observed to spend approximately one third to half the time measured attempting to discern how to distribute a GSP in order to chat with the facilitator. Section 3.3 details concepts that once leveraged could show gains in usability and reduce the learning curve by altering existing functionality and using social contexts with context-aware proactivity to aid in group formation.

3.2. Active Feedback on Current Context and Context Usage Reduces Distraction. After conducting the heuristics evaluation, providing proper feedback to the user in terms of how the system and application use context to support proactive functionality was considered to be an area in need of improvement. A discussion about the importance of proper feedback being given to the user when writing context-aware software and it's relationship to the Ubicomp concept of "fading-away into the background" is first given. Next, alterations made to give the user access to the current location identifier, and view the status of user group sessions quickly in the group manager is described. The changes made to provide proactive notification to the end-user about the start or end of a group interaction when chatting is reviewed. Finally, the problem of validating user-defined context data in a peer-to-peer system is then presented.

When developing software to proactively perform actions based on detecting a situation in the environment, proper feedback about how and why the software performed an action is important. A brief discussion about balancing proactivity and transparency [16] also posed similar questions about how to give a user proper feedback when using context-awareness. As part of making functionality so seamless that it does not require explicit user focus and "fades-away" into the background, sufficient feedback needs to be constructed so users still feel in control regardless of actions proactively being completed based on detected contexts. From Activity Theory it can be seen that if proper technique and construction of feedback to the user is used, even when at first it is distracting, eventually the action will be incorporated into a seamless cognitive operation as shown in figure 8. A few cases from testing that illustrated the importance of proper feedback to the user are discussed next.

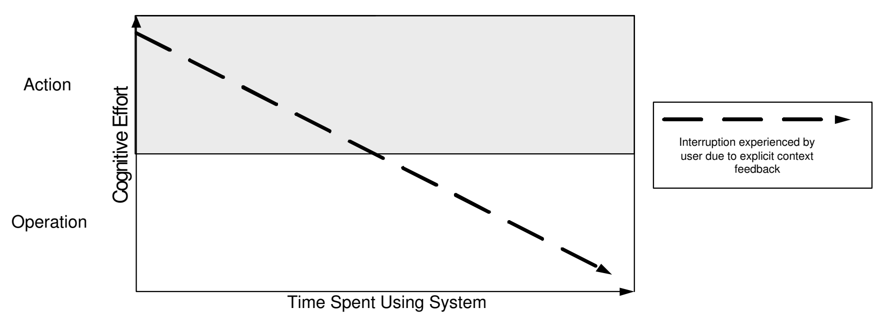

Figure 8. Over time processing feedback from an application takes less cognitive effort and becomes an operation instead of an action.

The initial version of the Group Manager, and Group Chat did not provide the user a means to see the current location as detected by the system. Without the ability to see the current location the end-user would not have all the necessary information at their finger-tips to determine why a group interaction disbanded and hence a chat ended. Although a user will have a sense of the current location in terms of their understanding of an area, explicit feedback is needed for cases where the boundaries for location as created by location beaconing may not completely match their perceptions. For example if a user is not aware that they had changed locations and they can see that a group should still be active due to time they may feel that an error had occurred with the application and misconstrue the situation. Giving the user the ability to see their current location and learn the location beacon topography is a means to eliminate this scenario.

Also the initial version of the Group Manager did not allow the user a means to easily discern the status of a User-group session. As described in chapter 4, section 2.3 a group session according to the system can be in a stopped, started, or active state. In version 1.0 of the group manager you could only tell if a group session denoted by a GSP had been stopped or started at a glance as denoted by the asterix next to the GID.

It was up to the user to review the status of the contexts captured by the GSP and figure out if the group session should be active. A task that couldn't be done very easily as you could not see the current location. Figure 9 shows the GUI for the initial and current versions of the Group Manager. In order to let the user know that the context for a group session is or is not met lights were placed next to the GSPs. Red for stopped, yellow for started but inactive, and green for active. In this manner the user can quickly glance at the Group Manager and see which groups are interacting in sessions, and hence can tell what the system is reading from the environment.

Usability testing scenarios were conducted in order to have users both enter and leave group sessions based on the context of a situation detected by the system. In the case of a chat session starting the users did not feel the need to have any more feedback than simply seeing the group show as active in the Group Manager, and display itself in the Group Chat. This

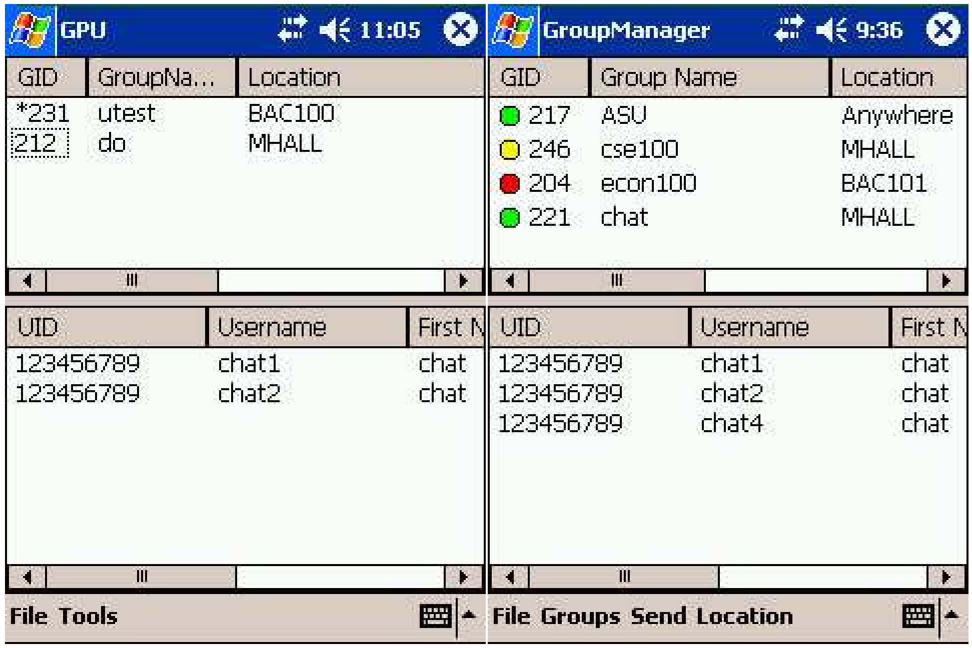

Figure 9. The old and new versions of the Group Manger interface. Newer version with enhanced interface to let the user understand the status of group interactions.

was most likely because users had full control over GSP distribution and hence full control over whom joined in the group. When a chat session ended due to a change in location or time however the users did not feel as comfortable. A user stated that they were not notified why they had been "kicked out" of a chat session. The fact that they felt kicked out implies that they felt a loss of control when the group session proactively ended itself and they didn't immediately know why. A message was already being displayed to the user to indicate that a chat session was ending and prompt them to determine if they wished to save the chat to a file. The message was expanded as shown in figure 10 to tell the end-user if the chat session was ending because they changed location, or the time for the session elapsed. In the case where locations were changed, the previous and current locations are included in the message.

Another instance uncovered by testing where a means needs to be devised in order to deliver proper feedback to a user anytime/anywhere resulted from having the users configure a GSP that was not in their current location. Most users automatically picked either what they imagined an abbreviation for "Murdock Hall 101" would be or entered in the name of

Figure 10. A message is displayed when a group chat session exits telling you which context changed.

the building verbatim. The application did not offer them any feedback in terms of if it was a valid location identifier, so users did not know if the group would actually start until they arrived at that location. Proper validation should be instituted when an end-user needs to specify values of social contexts that will be used to automate software in this circumstance, the problem is more straight-forward to solve with infrastructure support however for a truly anytime/anywhere solution it should work in ad-hoc situations such as the one explored by our prototype. The next section details findings from testing about how users feel social contexts about groups and users may be used to aid when discovering or configuring groups for use with applications.

3.3. User Group Sessions as a Context. The most difficult portion of working with groups in our prototype that was mentioned by users dealt with finding and distributing groups. Table 3 shows the reactions from three test users when asked the question "Was there anything about using the Group Manager utility you found confusing, or difficult when

User Response
Understanding how people connect to groups and join them.
I found it a bit difficult to send a group to another person.
I had a hard time adding users to the first group I created but, after that it was easier.

Table 3. What users found difficult about creating groups

User Response
Add a class schedule feature, if others are in the actual class, not just in the area they will show up.
Have public and private groups, allow users to see a list of public groups.
Use interests from user profile to locate, and join groups.
Provide notification when a new group is in the area and a link to join.

Table 4. User suggestions to improving working with groups

creating a group?". Users were then also asked about how the process of working with groups could be improved, suggestions they gave are listed in table 4. It can be seen that users mentioned using social contexts related to a group or user in order to locate, download, or join groups. Here the existence of groups and user background as sources of context that can be leveraged are discussed.

As seen in table 4 a statement was made that when entering a location it would be useful to see the set of currently active group sessions in the area and potentially join them. This indicates that users feel the interactions of groups in a given situation is a group-centric context which they wish the system and applications to utilize. Of course other information about the group such as the discussion topic would be used to discern if the user wanted to join in a chat session. Another example of how the existence of a group session in a location could be leveraged as a useful context would be an application for scheduling rooms and equipment as described in example 4.

Example 4 *When users meet at a location many times they will need access to specific tools, and potentially need to schedule another meeting. An application could be constructed to* *schedule a room based on contexts defined in a GSP. A time and location could be found using a GSP based on user preferences, room vacancy, and the availability of equipment. The group could then submit the request on the fly during their meeting in order to re-schedule the same room, or get suggestions for a room to use for their next meeting.*

From the above cases it can be seen that the activity of a group based on a situation is another context that could be explored and used by context-aware applications to support groups. This facilitates one of the user's perspectives of locating and joining groups from a group-centric point of view. The existence of group sessions for a given situation is available through examination of the contexts in a GSP. In order to leverage this as a context for applications using the social group model one could construct a service application. The service would query a set of GSPs and handle requests from applications to serve data about how, when, and why group sessions occur for a given situation.

3.4. Formal Group Membership as a Context, User Interests as a Means to Create Informal Group Membership. In addition to simply wanting the ability to see the activities of groups in a location it was expressed that viewing and leveraging social contexts about group members was desired. Primarily this was requested as a means to provide users with a user-centric means to locate and form groups to use for applications. Specific examples of information requested by users were class schedules, and interests. A class schedule is actually a means of determining the formal group memberships of a user. Interests although currently used by instant messengers could be used more proactively with context-awareness to create informal group memberships.

Group membership itself is seen as a social context that users may find useful when leveraged by applications. Currently all users in an area that are available to use the group chat are displayed in the Group Manager. In the case of a large area with over a hundred students the list of people to choose from could become hard to navigate. It was proposed that information such as the class schedule of a user could help limit the users that are displayed and can receive a GSP. So if you are in a CSE 100 class and defined a GSP for a group session to discuss that course only other people in the area whom are in a CSE 100 class would show up as able to receive the GSP.

User interests can be used by msn messenger as a means to locate contacts to interact with. Interests are not proactively used as a context by msn-messenger; they must be explicitly entered into a search form. User interests could be used proactively to facilitate both computer supported and in person interaction between users in an area. An example follows to illustrate this concept.

Example 5 *When traveling it is common to have long lay-over's at airports. Consider two individuals at an airport with long lay-over's for flights using the group chat application with their PDAs. When the two devices exchange UGPs and the group chat discovered that they have the interest "motorcycles" in common it could automatically create a GSP. The two could then chat over their PDAs and eventually perhaps decide to meet in person while they wait for their flights.*

In example 5 it was shown how an interest context in a UGP could be used to automate the creation of a GSP with the purpose of discussing motorcycles. In this example membership in a group based on interest in motorcycles is an example of an informal group membership being used to support group interaction. Another example to illustrate how informal group memberships can be formed to help users is given in example 6

Example 6 *Consider people travelling along an interstate it is common for people to drive in packs when making long roadtrips. An apple iPod can interface with a car stereo by plugging into a device that scans for a free FM frequency to broadcast on over short range. By placing music interests in a UGP along with the frequency you are transmitting on users in other cars could have their car stereo tuned in to listen to the same music.*

Commercial uses for forming device groups using interests and informal group memberships exist as well. Consider users with personal computing devices in a bookstore. A service could be built to mine interests in UGPs and perform statistic calculations on how often some occur. The statistics in turn could be used to aid in inventory management. Obviously using interests in this way carries major ramifications for privacy.

Contexts about a user stored in a UGP indicating group memberships can be used to aid in selecting people to send a group to. Also social contexts such as interest can be used to proactively automate forming groups between people when no formal group membership exists, and foster in-person interactions. Privacy concerns are touched upon next. Then in the next section future work and possible enhancements to the software prototype in order to enhance feedback on contexts to the end-user, and aid in forming groups using contexts in the GSP and UGP are detailed.

3.5. Privacy Ramifications. Each test-user mentioned their concerns about protecting the knowledge of their presence in an area, and membership in some groups private. Many of the contexts such as class schedules although highly useful need to be protected for obvious reasons. Similarly although users may wish to talk to a group about one of the interests in their profile, they may not wish the other user's to see all of their interests. This indicates that users need to be able to control somehow what and how much of the contexts in the GSP and UGP are able to be viewed by others.

4. Future Work

Possible solutions for providing better feedback to the end-user about how contexts are used, and expanding the use of group and user social contexts within our prototype is presented here. First considerations that need to be taken into account in order to aid users in distributing and obtaining GSPs is detailed. Next, a conceptual service for serving social contexts in peer-to-peer ad-hoc environments is described. Finally any additional considerations that concern desired future work are covered.

4.1. Distributing and Obtaining GSPs. The step that users found the most confusing about allowing a person to join in a chat group with them dealt with distributing the GSP. They also mentioned that they would like an easier means of obtaining GSPs so that they could talk to friends or meet new people in the same location as them. This was somewhat anticipated to be a desired addition to the prototype however it was desired to obtain the results of usability testing in order to receive user feedback on how obtaining GSPs would be expected to function. Here possible solutions to this problem that were developed based on feedback given by users is considered.

Users mentioned that they would like to choose who to distribute a GSP to upon creating the group. This follows a typical way that instant messengers build buddy lists. The challenge involved with providing this functionality is that there is no central server that builds up a list of users that can then be added based on social contexts such as name, and email address. Context based functionality could be added that uses social contexts such as email, and name to automatically distribute the GSP to a user when a matching profile is found in the current vicinity. The danger here comes from people that would spoof information in their UGPs to obtain access to groups that they did not belong to. In some cases due to co-location a user can physically see an individual to ensure that they have sent it to the right person however, depending on the expanse of a location this may not be possible.

As previously mentioned in section 3.3 when asked what improvement could be made in the process of joining groups, users wanted there to be some way to browse groups in an area, or automatically obtain groups based on social contexts such as membership in a class, or the discussion topic of a group. These contexts could be added to the UGP and GSP, which then would be used to filter and display or automate joining groups that are bound to a specific location or fit specific contexts in the profiles. This would potentially reduce the time needed to learn how to work with groups, and create informal group memberships to foster spontaneous social interactions between users when in a location such as a college campus.

Privacy also needs to be factored into the system in order to allow users the ability to conceal their presence in an area, and limit exposure of some social contexts in their profiles. It was suggested that groups be segmented into two basic types, public and private. In this case only "public" groups could be viewed and downloaded by browsing for groups or joining them based on social contexts. Users would then be allowed to create and distribute "private" groups between each other that are not advertised by selecting specific individuals that they know and trust. Test feedback clearly indicates that users desire a solution for allowing them to browse GSPs and form groups based on social contexts in order to improve usability. Section 4.2 describes the concept of a service for providing this functionality.

4.2. A Social Context Distribution Service. This section describes the concept for a peer-to-peer service for sharing and distributing social context and GSPs. The creation of a Social Context Distribution Service(SCDS) would be used to enhance the abilities of applications to use the device group as a tool. First how using an SCDS to support the distributing and browsing of GSPs would allow for a new application and enhance the group chat is covered. Next a means to help support the process of user's creating GSPs that are bound to a specific location is described.

4.3. Browsing and Distributing Profiles. As part of creating a SCDS a peer-to-peer algorithm to distribute profiles based on social contexts could be investigated. Additionally the SCDS could support explicit querying of social contexts. It could then serve social context anytime/anywhere or distribute social contexts at a location. Part of distributing social context would include the existence of GSPs as a context.

The SCDS would then be used by stand-alone applications that wish to use social-contexts via profiles and as a means to locate GSPs to form device groups. New means to support applications to foster group interaction can be created this way. For example the set of groups tied to a location is indeed a context that can then be leveraged by applications other than chat. Consider the task of scheduling a room for a team meeting.

Groups that have already reserved that room for a specific time are an important social context that the group may wish to have access to. With a service that supports the ability to distribute GSPs at a given location, a scheduling application could be developed that allowed the group to schedule their meeting while in that location. To do this a group member would create a GSP and upload it to the SCDS in that room for a time that is not used by another GSP. Additionally tools that are needed by the group could be placed in the GSP and reserved for the group as well.

Returning a list of GSPs that is bound to a location is one example of querying for a list of group sessions based on a specific social context. The SCDS should support querying for GSPs for any social context that is defined in a GSP. Applications could then make calls to the SCDS in order to find GSPs for whatever contexts they use to proactively form group sessions. A similar service could be provided by the SCDS for browsing and locating users from UGPs.

Functionality could be supported by the SCDS to search through UGPs it has received and return a list of UGPs that meet the search parameters. This could be used by the group manager to form informal group memberships on-the-fly in order to create opportunistic introductions between people. In this case a protocol would need to be developed that allowed the user to refuse taking part in a session without other users discovering that they declined.

It could also be used to filter through the list of UGPs in an area to find users based on formal group memberships.

The SCDS could also allow the creation of public and private groups if desired. In order to facilitate this a group marked as "public" could be uploaded to any available SCDS running in an area or on a device. Groups marked as "private" upon creation would then only be distributed explicitly by the end-user to other trusted parties. In the case of public groups though the user will need a means to control turning on/off having GSPs uploaded to a SCDS in consideration of bandwidth and energy resources.

A portable device running an SCDS to distribute GSPs based on social contexts in their user profile such as classes, and interests allows the Group Chat to be a means of meeting people with similar courses or hobbies. The knowledge of a person with similar interests in the area and the ability to form chat groups based on informal membership criteria could prompt people to interact in person as given in example 5. The use of an SCDS also provides users with the requested ability to form and interact with both public and private groups.

4.4. Providing Validation for User-defined Contexts. An SCDS could also be set up to collect location IDs, and canonical names for them from GSPs that are created or sent to it. This could be facilitated by allowing the user to specify a canonical name for the location tied to a GSP, and store it in the GSP. The SCDS could then map canonical names to location IDs and create more valid location IDs based on what people commonly want to use. The list of locations collected at the SCDS could then be used to validate, suggest, or browse locations when a user creates a GSP.

4.5. Additional Considerations. Some additional things to consider if a SCDS is built is the need for a management application that interfaces with it in order to allow administration of the service. Also a framework containing a set of re-usable controls should be made to interface with the SCDS and SAEG that would then be integrated into applications in order to provide functionality to work with social context through the application's interface. For example the controls could be integrated into the chat application to provide the ability to add groups to chat with from within the chat. Also it may be useful to allow the SCDS to tie itself into infrastructure to provide access to social-contexts stored in other locations.

5. Conclusion

The results of evaluating the Group Manager and Group Chat applications through usability testing were presented in this chapter. Prior to users scenario testing the applications a Heuristics evaluation was performed. Next, 4 users [13] were recruited to test the application by completing three different scenarios. A one-on-one test creating group sessions with the Group Manager, and chatting with the author using the Group Chat. Creating group sessions and chatting with the other three users as a group. Utilizing the software as group sessions ended due to changes in location or time.

It was determined that the user needed better feedback on how the applications were using context to aid them. Although context-awareness is used to fade system functions into the background in some cases actively giving feedback to the user is what allows the application to support the user without causing distraction. Furthermore if feedback needs to be actively provided to the user and it is done properly even if it is distracting at first eventually over time it will be fluidly processed. In Activity Theory terminology it will go from being an action to an operation as seen in figure 8.

Applications were improved by giving the user more feedback on the current context being used by the applications. Improvements as shown in figures 9 and 10 included: Adding functionality that allowed applications to report the current location as seen by the SAEG. The ability to discern the context at a glance by seeing icons indicating the stopped, started, and active status of group sessions from the Group Manager. Group creation was enhanced by allowing the user to push a button and fill in the location for a GSP with the current location preventing errors. Finally when a group involved in a chat session exited due to context changes the Group Chat application now provides a message describing which context change caused the group session to quit.

During the first scenario users were timed while creating a GSP and distributing it to the author to assess the learning curve for creating and forming a group. The average time was 10 minutes and 36 seconds, and users were observed to have spent about a third to half of the time completing the task on distributing the GSP. The fact that GSP distribution took so long indicated that improvements should strive to reduce the learning curve for that step, feedback from users shown in table 3 confirmed this. Users were asked how this could be improved and made the suggestions in table 4.

In order to improve forming groups users wanted the ability to see the existenc e of a group session in an area so that they could review it's GSP and possibly join it. The idea of public and private group sessions was also requested. Social contexts about the users such as class schedules and interests where then requested to be used as a means to either limit the number of users whom could receive a GSP, or create informal group memberships and automatically create group sessions.

Future work was described that would address user requests and improve system and application usability. This would include the development of a new service called the Social Context Distribution Service. The service could be used to store GSPs for the purpose of allowing users to browse them as public group sessions. A private group session would then be a GSP marked as private and distributed using the Group Manager to trusted users. The service could also proactively use the information in a UGP to find and suggest existing GSPs, or create them on the fly to aid users in meeting one another.

After development and integration of the Group Chat with the SCDS has been completed, usability testing should be conducted again. Timing should be taken to assess improvements in usability in terms of creating and forming groups. Also a distraction matrix could be constructed to demonstrate improvements from feedback to the user, and proactive formation of groups based on social contexts in the UGP and GSP. In the next chapter an overview is presented along with conclusions drawn from the conducting of this research.

CHAPTER 6

Conclusion

1. Purpose

Future computing environments will strive to support users in all aspects of life from work, to play on an individual and group level. By giving devices the ability to read and interpret the current contexts of a situation and embedding them throughout the environment computing usability will reach new levels of fluidity. Devices will not only be placed in institutions but also be completely woven into the fabric of life through wearable computing. Thus systems running on devices worn by users cannot always rely on infrastructure support and must provide as close to an equal measure of functionality when in isolation.

The specific focus of this work was to seek further research results involving supporting social groups of people in Ubicomp environments. Devices throughout a Ubicomp environment should create the perception that they understand the purpose of groups as the user does from a social standpoint. Thus devices should network together to assist users collaborating in a manner that does not cause interruption. In some cases this means proactively notifying users to keep them informed on system status and allow them control over what the systems are doing. Unless the user is empowered as such, technologies to support individuals and groups of users will never reach wide-spread adoption.

2. A Model for Developing Software to Support Social Groups

The model presented in this section was detailed in chapter 3 and is a conceptual model developed for the purpose of supporting groups of users in a Ubicomp Environment. It was based on the definition of social groups as given in section 1 and found in [12]: "A number of individuals, defined by formal or informal criteria of membership who share a feeling of unity or are bound together in relatively stable patterns of interaction". A model for supporting social groups using computation must support the users perceptions of membership, and facilitate group members interactions with one-another for the appropriate durations and situations. Additionally it should ensure a general means to support the mediating relationships affecting the subject, and community entities in Activity Theory through context-awareness.

Two levels of groups were defined in the model, the User-group (social group) and the device group. The subjects or individual devices of each respective group are aggregated into a group structure based on the two general contexts defined in section 4 of chapter 3. The User-group forms to work on an activity based on social factors defined as Social Contexts.The Device-group forms based on the Device Context, the Device Context is the state of the device as it receives input from the user and the environment.

The relationship between the two was shown in figure 2. It can be seen that the device group must form to support the activities of the User-group. Thus the Social-contexts important to the User-group must somehow be represented in a form that allows them to become part of the Device-context in order to network devices together proactively to support the needed actions. To do this profiles are used to place social contexts in a form that a device can access. Two profiles exist in the model: The User-group profile(UGP) that contains data important to representing social contexts important to the user such as their name, and preferences. A Group-session profile(GSP) used to contain the social contexts used to describe the social situations that cause a device group to form and disband in order to properly facilitate group activity.

3. A Prototype Software System to Support Social Group Communication

As stated previously, in order to determine the feasibility of building software based on the group model described in section 2 a prototype group collaboration system featuring chat and file exchange was built. The software services and applications that the prototype comprises of are described below:

1. **Situation-Aware Ephemeral Group(SAEG) Service -** Processes GSPs to network devices together peer-to-peer into an asynchronous group under a common group identifier. Messages are then sent through the SAEG using the group identifier and received by each device in the group, several groups can be supported at once.

2. **User Profile Distribution(UPD) Service -** A peer-to-peer service that distributes the UGP profile for a device to other devices running the same service. A use of this service is announcing the presence of a specific device. Time outs are used to validate if a device is still in range.

3. **Group Manager Application -** An application used to author GSPs, store them and let the user decide what devices to distribute the GSP to, and hence what users are supported by a device group.

4. **Group Chat Application -** Supports group collaboration through text messaging, and file exchange. Chat Groups are formed based on GSPs and are analogous to a chat channel in an application such as IRC.

As part of exploring the support of social groups in a Ubicomp environment usability testing was conducted from which the primary conceptual contributions of this thesis were obtained.

4. Results of Usability Testing

Usability testing included a heuristics evaluation, and scenario testing with 4 users [13]. Scenarios tested: Creating a GSP to form a chat Group with the facilitator. Creating a GSP to form a chat group with three other users. Using the Group Chat as context-awareness caused group chats to end due to changes in location or time. Users were timed for the first scenario and were found to have taken an average of 10 minutes and 36 seconds to create a GSP and form a group using the Group manager. Most of that time was spent trying to form the group by sending the GSP to another device indicating that proper concepts still remained to be uncovered for improving usability in that respect. User feedback confirmed that forming groups was an area they felt was in need of improvement.

From the final test where groups disbanded due to changes in context it was found that users wanted more feedback and fine-grained control over how groups reacted to context. This was evidenced by users commenting on being "kicked" out of the groups, implying that they felt a loss of control and were frustrated. Three major concepts were contributed by this research after completing testing as presented in the next section.

5. Contributions

The primary findings of this research are as follows:

1. Keeping the user informed through active feedback about context and how it is used by an application to support group interactions is part of minimizing distraction.

2. The existence of social group collaborative sessions is a context that applications can leverage to support social groups.

3. The formal and informal group memberships of a user can be used by context-aware software to support social group interaction.

Supporting users while reducing distraction does not infer that they are not notified or queried to supply feedback on actions being taken. In particular when supporting group interaction using context-awareness it is still necessary to give the user proper information and control over software actions in order to keep them from becoming frustrated. In many group interactions there are a myriad of social implications depending on the action taken by an individual, and it may not be possible to enumerate each case and have software properly handle each without feedback from the user. The cases of forming and disbanding groups due to changes in context were explored in this research.

In the case of joining groups it was noted that users did find the process of distributing the profile cumbersome and requested a means to find and join publicly active groups in the area. However once a GSP was distributed and the users were in a group they made no comment on the fact that a group had activated due to time and location. This is most likely because they had been given ample control over controlling the membership of the group and they felt like chatting. In the case of leaving groups due to changes in time or location they did not feel that they had an ample understanding of why a group had quit, or ample control over their membership in the group. Users felt as if they had been "kicked out" of the groups, a comment indicating that completely automating the end of a group session would lead to agitation and frustration, and hence distraction. Proper feedback about what contextual change triggered the end of a group interaction has been added into the group chat but, user requests to enable delaying the disbanding of the group has not yet been included.

The existence of a group's sessions per a given location or situation is a useful piece of contextual information that's use should be explored. This was showcased in this research primarily because test users wanted the ability to find public group sessions that they may take part in when entering a location. However, other uses exist if the knowledge of a group session is made available such as scheduling rooms, and resources as demonstrated in example 4 from chapter 5. Another facet of this concept is that users expect to be able to work with social groups through the application interface in a group-centric means based on contexts that are properties of the group as a whole. Properties of the group such as the current contents of it's membership and the purpose of the group are important when making decisions about joining the group. The current prototype does not support the concept of a public browse able group session, this is however an interesting concept for further development and exploration.

The concept of group memberships as indicated by the user and establishments are also a useful context to be leveraged. Information such as a student's course schedule indicates their memberships in class groups. The knowledge of an individual's inclusion in a group may meet the needs of another group seeking to add a member with a specific skill, or a user looking to form a group for a common purpose. In chapter 3 example 3 a study group was looking for someone with the role of "physics tutor". In this case the role of tutor may also indicate their membership as an employee in a group of tutors employed by the college. Example 5 from chapter 5 demonstrated the use of an interest in motorcycles as a means to indicate and leverage informal group membership that put two people in contact with each other in order to discuss their shared interest. Using group membership as a context to find potential members and form new groups is a user-centric means of forming groups from the perspective that contexts added by the user are leveraged. The fact is that as people we expect to work with groups based on contexts added because of the group as an entity, or because of contexts from a group member's profile.

6. Future Work

The focus of future research efforts would be to further explore the concepts that were contributed and detailed in section 5. To do this an application framework along with the Social Context Distribution Service proposed in section 4.2 can be constructed. The main goal of developing an application framework would be to create re-usable tools that would allow rapid-prototyping and development of applications that wish to support groups per the Social Group model. Tools would interface with the existing SAEG and UPD services along with the SCDS to support providing proper feedback, and allow users to work with groups using both group-centric and user-centric methods that were requested by test users.

The purpose of developing a SCDS would mainly be to allow group sessions to be formed or located based on social contexts. This would include locating groups based on the contexts that describe a group session, and finding members to form a group with based on social contexts that are contained in a UGP. The SCDS would do this by supporting peer-to-peer functions to search through GSPs for specific contexts and then download them; it would support a similar function on UGPs. A list of UGPs, or GSPs would be returned that matched the specified search parameters. This would also allow for the creation of public and private group sessions. Public GSPs would be uploaded to an SCDS while private would only be distributed explicitly by the end user. Similar to the research in [5] the SCDS could be used to refine the contents of the UGP kept by the owner of the device running the SCDS.

7. Closing Statements

Future computing environments will support us as individuals and groups as we work and play. In order to achieve this new levels of Human Computer Interaction will be needed in order to build the perception that computers can store and process a view of the world like the user's and assist them as they go about daily life. The importance of proper Usability studies will grow as proactive systems take on assisting users to carry out increasingly more socially sensitive tasks. It is also the author's opinion that observing how people complete tasks using ehtnomethodelogical studies and similar sociological and psychological techniques will need to have more emphasis in software engineering processes in order to understand how and why proactive context-aware applications assist us in Ubicomp environments.

REFERENCES

[1] S. O. A. Ferscha, C. Holzmann. Context awareness for group interaction support. *Mobiwac*, pages 88–97, 2004.

[2] G. D. Abowd, C. G. Atkenson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wireless Networks*, 3(5):421–433, 1997.

[3] J. Fleming. User testing. *www.ahref.com*, Mar 2006.

[4] D. Garlan, D. P. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 1(2):22–31, 2002.

[5] J. Gips and A. S. Pentland. Mapping human networks. In *Percom*, 2006.

[6] J. Grudin. Group dynamics and ubiquitous computing. *Communications of the ACM*, 45(12):74–78, Dec 2002.

[7] C. A. Halverson. Activity theory and distributed cognition: What does cscw need to do with theories? *Journal of Computer Supported Cooperative Work*, 2001.

[8] E. Hutchings. *Cognition in the Wild.* The MIT Press, 1995.

[9] E. Isaacs, A. Walendowski, and D. Ranganathan. Mobile instant messaging through hubbub. *Communications of the ACM*, 45(9):68–72, 2002.

[10] D. A. K. Understanding and using context. *Personal and Ubiquitous Computing*, 5, 2001.

[11] K. Kuutti. *Activity Theory as a Potential Framework for Human-Computer Interaction Research.* The MIT Press, 1996.

[12] G. Marshall and et al. *The Concise Oxford Dictionary of Sociology.* Oxford University Press, Great Britain, 1994.

[13] J. Nielsen. Why you only need to test with 5 users. *Usetit.com*, Mar 2006.

[14] A. Ranganathan, R. H. Campbell, A. Ravi, and A. Mahajan. Conchat: A context-aware chat program. *IEEE Pervasive Computing*, 1(3):51–57, 2002.

[15] M. Reseach. Microsoft easy living. *Microsoft Easyliving Home Page*, Mar. 2006.

[16] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17, Aug. 2001.

[17] A. Schmidt, M. Beigl, and H. W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

[18] L. Schuman. *Plans and Situated Actions.* Cambridge University Press, 1987.

[19] M. e. a. Terry. Social net: Using patterns of physical proximity over time to infer shared interests. In *CHI*, 2002.

[20] M. L. W. Newman, M. Eldridge. Pepys: Generating autobiographies by automatic tracking. In *Second European Conference on Computer Supported Cooperative Work*, 1991.

[21] B. Wang, J. Bodily, and S. Gupta. Supporting persistent social groups in ubiquitous computing environments using context-aware ephemeral group service. In *Percom*, 2004.

[22] R. Want, A. Hopper, V. Falco, and J. Gibbons. The active badge location system. *Olivetti Research Ltd. (ORL)*, 1992.

[23] M. Weiser. The computer of the twenty-first century. *Scientific American*, 265(3):66–75, Sept. 1991.

[24] P. Werle, F. Kilander, M. Jonsson, P. Lonnqvist, and C. G. Jansson. A ubiquitous service environmnet with active documents for teamwork support. In *Ubicomp 2001*, 2001.

APPENDIX A

TEST INSTRUMENTS

1. Usability Test Task Scripts

Task: One on One

1. You have just arrived and sat down at your intro to macro-economics class in a large lecture hall in BAC. Unfortunately you got delayed by traffic and have arrived a few minutes late. You see a good friend of yours sitting on the other side of the hall. After the first class you had talked about using your PDAs to chat in the event that one of you arrived late, however you never took the time to set up a chat group. Your friend sees you and motions to his PDA.

2. You remember that you have a CSE 100 class at 1:00 p.m. in the "Murdock Lecture Hall" building tomorrow. You decide to create a group profile for use in this class in case you or your friend are not seated next to each other and you want to discuss the material being covered in the lecture.

Task: Chatting as a Group

1. You and your friends are taking CSE 100 at ASU in the "Murdock Hall" building from 1-1:50 p.m. The class is taught in a large lecture hall and you dont always get to sit together, but you would like to be able to chat using your PDAs during class to discuss what the instructor is reviewing. It is the first day of class you and another friend that carpool have arrived and taken seats. You see a couple of your friends on the other side of the hall and decide you want to start a chat.

Task: Leaving Groups

1. You and your three friends are finishing up your CSE100 class that is being taught in the Murdock Hall. You are all enrolled in an Economics class during the next class period being taught in room 101 in the BAC building. You have already created profiles for these classes; the class in Murdock Hall is letting out.

2. Pre-Test Questionnaire

1. Have you ever used a Pocket PC device before?
2. Have you ever used an application designed to work over an ad-hoc network?
3. About how much time have you spent using chat applications?
4. How familiar are you with research in the area of Ubiquitous computing?
5. How familiar are you with research in the area of Context-Aware computing?
6. What activities do you participate in with groups of people?
7. What software do you use when participating with groups?

3. One on One Questionnaire

1. Was there anything about using the Group Manager utility you found confusing, or difficult when creating groups?
2. What was easy about using the Group Manager utility when creating groups?
3. Besides group name, location, and a time span what other information could be added or used to further define a group for the purposes of activating it, joining it, leaving it, deactivating it, locating it?

4. Was there anything difficult or confusing about sending or accepting a group?

5. What was easy to understand, or perform when sending or accepting a group?

6. What features would you add in order to make creating, locating, obtaining, and then managing groups once obtained or created easier or more useful?

7. What features would you change in order to make creating, locating, obtaining, and then managing groups once obtained or created easier or more useful?

8. What was difficult or hard to understand about using the GroupTextChat application?

9. What was easy to use or understand about the GroupTextChat application?

10. What features would you add to the GroupTextChat we used today in order to make it easier to use?

11. What features would you change in the GroupTextChat used today in order to improve it?

4. Group Test Questionnaire

Task: Chatting as a Group

1. Was there anything about using the Group Manager utility you found confusing, or difficult when creating, sending, or accepting groups?

2. What was easy about using the Group Manager utility when creating, sending, or accepting groups?

3. Besides group name, location, and a time span what other information could be added or used to further define a group for the purposes of activating it, joining it, leaving it, deactivating it, locating it?

4. What was difficult or hard to understand about using the GroupTextChat application?
5. What was easy to use or understand about the GroupTextChat application?
6. What features would you add to the GroupTextChat we used today in order to make it easier to use?
7. What features would you change in the GroupTextChat used today in order to improve it?

Task: Leaving Groups

1. What was easy to use or understand about using the chat application when you are leaving a group due to changes in time, and/or location?
2. What was difficult to use or understand about using the chat application when you are leaving a chat group due to changes in time, and/or location?
3. What features would you add to the chat in order to make using it when chat groups exit from time and/or location changes easier?
4. Are there any features you would change to make using the chat easier when a group exits from time and/or location changes?
5. What existing applications could you see leveraging location awareness and context-awareness to add beneficial functionality?
6. What new applications can you envision using location awareness and context-awareness?

5. Obtaining and Installing the Prototype

The software prototype described in chapter 4 that was used during usability testing can be downloaded by visiting the following webpage:

http://shamir.eas.asu.edu/ mcn/GCPrototype.html

The author is not responsible for supporting the prototype software. It is intended to be downloaded and used as is. For those with access to Embedded Visual Studio C++ (EVC) the source code is available in a file named **GCPrototypeCode.zip**. For those that intend to use the platform as described next, the executables can be downloaded in the file named **GCPrototypeExec.zip**.

5.1. Prototype Platform. All software was designed and built to deploy on the Dell Axim x5 PDA running Microsoft Pocket PC Version 4.2. The specific hardware specifications of the PDAs used are listed below:

1. **Processor Type:** Intel XScale
2. **Processor Speed:** 400MHZ
3. **RAM Size:** 64 MB
4. **ROM Size:** 48 MB
5. **ROM Version:** A05
6. **Wireless Card:** Dell TrueMobile 1180

All software development was performed using the Embedded Visual C++ 4.0 IDE.

5.2. GCPrototypeCode.zip File Contents. The source code for the Group Chat prototype system is compressed within the GCPrototypeCode.zip file. This section contains a description of the projects that are contained within the file in two directories: SystemApps and UserApps. The directory named SystemApps contains the following applications:

1. RCSM Object Request Broker (R-ORB) - The R-ORB is responsible for context acquisition and acts as a context-sensitive object broker. For more information see: http://dpse.asu.edu/rcsm/RCSM-software.html

2. R-ORB CFG - A utility application that creates a configuration file needed to execute the R-ORB.

3. R-ORB Shutdown - A utility that shuts down the R-ORB once it is executing.

4. R-ORB_loc_change_sim - A version of the R-ORB that simulates a location change occurring 2 minutes after it is executed. The starting location will be read from the R-ORB config file. After 2 minutes the location will switch to "BAC101".

5. Situation Aware Ephemeral Group (SAEG) Manager - A service used to form device groups by creating non-deterministic process groups. Device groups are formed based on contexts found in a GSP. The SAEG returns a unique identifier for the GSP and allows applications to send messages to all devices in the group using the group identifier. More details about the service were published in [21].

6. User Profile Distribution (UPD) - Used to distribute and receive UGPs between devices running the UPD. Announces the arrival of a user in an area, and allows applications to query the set of nearby users.

7. UPD Shutdown - Used to turn off the UPD on a device.

Within the UserApps directory you will find:

1. CreateUP - A utility for editing the UGP on a device.

2. GroupManager - An application for managing the creation, storage, and distribution of GSPs. The application is more fully documented in chapter 4 section 2.3.

3. GroupTextChat - The collaborative group software used to exchange text messages and files between group members. More documentation about the GroupTextChat application is found in chapter 4 section 2.6.

4. RCSM - The RCSM application acts as a means to bootstrap system software needed to use the Group Manager and Group Chat applications. It launches a process to start the R-ORB, SAEG, and UPD. Both the Group Chat and Group Manager will test to see if the R-ORB is running, and if need be call the RCSM app before displaying the GUI.

5.3. GCPrototypeExec.zip File Contents. For those wishing to use the same platform as the author, it may be easier to copy the binary executables onto the devices. The GCPrototypeExec.zip contains two main directories: SystemApps, and UserApps. Once unpacked in the SystemApps directory you will find:

1. /RORB_loc_change/rorb.exe - A version of the R-ORB that simulates a location change occurring 2 minutes after it is executed. The starting location will be read from the R-ORB config file. After 2 minutes the location will switch to "BAC101".

2. config_rorb.exe - A utility application that creates a configuration file needed to execute the R-ORB.

3. rorb.exe - The RCSM Object Request Broker (R-ORB) is responsible for context acquisition and acts as a context-sensitive object broker. For more information see: http://dpse.asu.edu/rcsm/RCSM-software.html

4. rorb_shutdown.exe - A utility that shuts down the R-ORB once it is executing.

5. saeg.exe - The Situation Aware Ephemeral Group (SAEG) service used to form device groups by creating asynchronous process groups. Device groups are formed based on

contexts found in a GSP. The SAEG returns a unique identifier for the GSP and allows applications to send messages to all devices in the group using the group identifier. More details about the service were published in [21].

6. UPD_shutdown.exe - Used to turn off the UPD on a device.

7. UserProfileDistribution.exe - Used to distribute and receive UGPs between devices running the UPD. Announces the arrival of a user in an area, and allows applications to query the set of nearby users.

Inside the UserApps directory you will find the following applications:

1. CreateUP.exe - A utility for editing the UGP on a device.

2. GroupManager.exe - An application for managing the creation, storage, and distribution of GSPs. The application is more fully documented in chapter 4 section 2.3.

3. GroupTextChat.exe - The collaborative group software used to exchange text messages and files between group members. More documentation about the GroupTextChat application is found in chapter 4 section 2.6.

4. RCSM.exe - The RCSM application acts as a means to bootstrap system software needed to use the Group Manager and Group Chat applications. It launches a process to start the R-ORB, SAEG, and UPD. Both the Group Chat and Group Manager will test to see if the R-ORB is running, and if need be call the RCSM app before displaying the GUI.

5.4. Before Prototype Installation. Previous to prototype installation the following steps must be taken using the Pocket PC:

1. Navigate to the "My Device" directory - Navigate to the root directory of the device using the "File Explorer" program.

2. Create the RCSM directory - Create a directory named "RCSM".

3. Create the App and Data directory - Navigate to the newly created "RCSM" directory and create two sub-directories; one named "App", and another named "Data"

At this point refer to the appropriate guide given below. If you are using EVC to install the prototype refer to sub-section 5.5. For directions on prototype installation by copying the executables to a PDA see sub-section 5.6. Directions for starting the Group Manager and Group Chat follow in sub-section 5.7.

5.5. Installing the Prototype Using EVC. Installation using EVC is very straight-forward. Depending on the platform you are building for, you may be required to download packages from Microsoft support online.

1. Sync the PDA - Sync the PDA up with the laptop by placing the PDA in it's cradle and running activesync.

2. Open Projects - Open up the projects in the zip file. Every project must be built and installed in order for the Group Chat prototype to work. Opening and installing them one at a time is probably preferable based on the number of projects. You should build only one version of the R-ORB, depending on if you want a location change simulated or not.

3. Configure the Project - The project should have the current build configuration set to release for the device you intend to use. For the platform described in chapter 4 section 1 the active configuration was: WCE Configuration = "POCKET PC 2003", Active Configuration = "Win32 (WCE ARMV4) Release", Default Device = "Pocket PC 2003 Device".

4. Build the Project - Provided you have created the proper directory structures the project will build. Projects unzipped in the UserApp directory will have the executable placed in the /RCSM directory. Projects unzipped from the SystemApps directory are placed in the /RCSM/App directory.

5.6. Installing the Prototype by Copying Executables. If you are using the platform described in chapter 4 section 1 you can install the Group Chat prototype by copying the executables onto the PDA. This requires two steps; Copy every executable that was unzipped in the UserApps directory into the /RCSM directory on the PDA. Copy executables from the SystemApps directory into the /RCSM/App directory on the PDA. You will need to copy only one version of the R-ORB, depending on if you want to simulate a location change.

5.7. Executing the Prototype. The following steps must be taken in order to use the group chat prototype to chat and share files: Configure the R-ORB. Create a UGP. Create a GSP. Distribute the GSP. Start the Group Text Chat application.

To configure the R-ORB you must run the config_rorb application found in the /RCSM/App directory. If you do not wish to use a location when experimenting with the Group Chat choose the "Basic" option and select "Apply". If you wish to use a location when experimenting with the Group Chat, fill in the location you wish to use in the text box labeled "Loc". Select the "Context Server (Loc Only)" option, and select "Apply". A config file for the R-ORB will be created in the /RCSM/Data/ directory.

The next step is to create a UGP. To do this execute the CreateUP application that is found in the /RCSM directory of the PDA. Fill in the username, first and last names, email, and ID then select "Save". A UGP for use by the UPD will be created and stored in the /RCSM/Data/ directory.

Now that a UGP has been created you can run the Group Manager to create a GSP. The directions for creating and distributing a GSP as given in chapter 4 section 2.4 are repeated below:

Creating a Group Session Using the GM - Creating group sessions is done using the GM. When creating a group session the user will first be asked to configure a group name, and a location used to activate the group. By default the current location is selected and displayed. The user can choose to change the location to a new one which they know the identifier for, or select an "Anywhere" location.

The next step involves configuring a start and end time for the group interaction represented by the GSP. When the user first views the interface the current time is used to specify the start time, and the end time is configured to be an hour later. The user can then set the start and end times to indicate the situation they wish, or specify an "Anytime" context. Upon completion on specifying the time the GSP is sent to the SAEG and used to place the group in the started, or active state.

Distributing and Receiving a Group - In order to distribute a profile after creation the user must select the profile and then select the "Send Group" option which is under the "Send" menu. Next the end-user is allowed to select one or many nearby users in order to send a GSP to. A GSP for a group is then sent over the network to the GM that is running on the remote-user's PDA.

Receiving a group is very straight-forward. when another user desires to send a group across the network to you a message will appear asking if you would like to review the profile. If the end-user possibly wishes to join the group they can review the GSP for the group and accept or cancel adding their device to the group. When accepting a GSP the device is not automatically put into the group, instead the user must start the group if they really want to join it.

5.8. Starting the Chat. Now that you have created and distributed a GSP you should be ready to use the Group Chat application to chat and exchange files. Remember that your chat group will only appear in the Group Chat provided that the context for starting the group interaction from your GSP is true. An example of an active Group Chat is shown in figure 11.

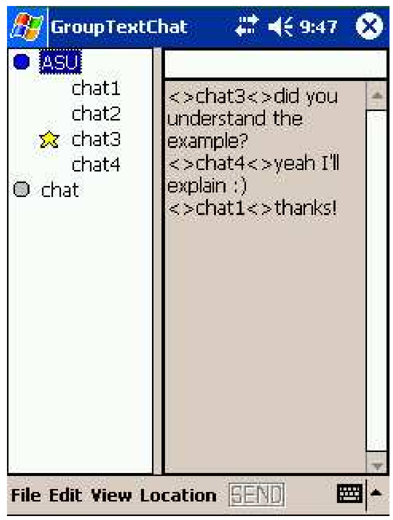

Figure 11. Screen-shot of the prototype GroupChat application.

You can easily determine if a GSP is active by checking the GSP in the Group Manager as seen in figure 12. A green light will appear next to the GSP if it is active. Yellow indicates that the GSP is started but, the context is not yet correct for the GSP to activate. A red light indicates that a GSP is stopped and it will never activate (Start and Stop are in the "Groups" menu). If you are currently chatting with a group and the context indicates that the chat interaction is over you will see a message as shown in figure 13.

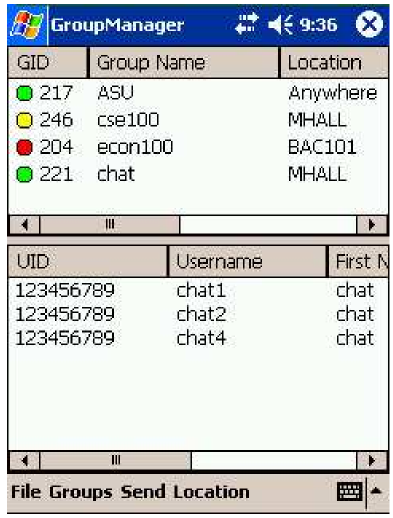

Figure 12. Screen-shot of the prototype Group Manager application. Lights next to a GSP indicate the state of a group interaction.

Figure 13. A message is displayed when a group chat session exits telling you which context changed.