This is the author's version of a work that was submitted/accepted for publication in the following source:


This file was downloaded from: http://eprints.qut.edu.au/46885/

© Copyright 2011 ACM

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. OZCHI ’11, Nov 28 – Dec 2, 2011, Canberra, Australia Copyright © 2011 ACM 978-1-4503-1090-1/11/11... $10.00

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:
ABSTRACT
The increasing capability of mobile devices and social networks to gather contextual and social data has led to increased interest in context-aware computing for mobile applications. This paper explores ways of reconciling two different viewpoints of context, representational and interactional, that have arisen respectively from technical and social science perspectives on context-aware computing. Through a case study in agile ridesharing, the importance of dynamic context control, historical context and broader context is discussed. We build upon earlier work that has sought to address the divide by further explicating the problem in the mobile context and expanding on the design approaches.

Author Keywords
Context-aware computing, Interaction, Representation, Dynamic ridesharing systems

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Early context-aware computing applications sought to take advantage of sensor information in order to better represent the context of the computing device and its user. Initially, sensors provided data on various physical quantities such as location and temperature and context–aware devices sought to make sense of disparate data through integration and computation of values. Applications were envisaged to be smart homes, phones that knew when not to ring. From a technical viewpoint, context referred to entities that could be represented and possibly used in computations, and thus this view is referred to as a representational view of context (Dey 2001).

The expansion of mobile devices, sensor networks and social networks such as Facebook and Twitter, has brought forward more kinds of data that can be mined in order to represent context. However a social and interactional view of context has found the technical aspiration to compute context as problematic, particularly when the aim is to advise or control applications that directly affect people’s activities. From a social perspective, context is a dynamic and emergent property that arises out of activity (Dourish 2004). Applications such as a phone with the intelligence to know when not to ring are doomed to fail because they will inevitably fail to interpret complex emergent social situations and make the right decisions (Brown and Randell 2004).

This paper demonstrates through design how to work from these two viewpoints in order to design a mobile application that uses context information effectively. We also introduce the importance of dynamic context control, historical context and broader context as necessary aspects of mobile context needed in ridesharing applications.

DESIGNING FOR CONTEXT-AWARE COMPUTING
Given the interactional perspective, Brown and Randall (Brown and Randell 2004) suggested the following design strategies for context aware technologies – That (i) Technology provide context to users through simple structures (such as. caller ID) allowing users to make sense of that contextual information themselves; (ii) Context be used defensively, in such a way that incorrect inferences will not be a serious inconvenience to users. (iii) Technology focuses on communicating context, rather than attempting to compute it.

Oulasvirta et al. (2005) concluded that a balance can be found between constructivism and realism approaches by making automatic actions based on sensor data negotiateable and repairable. Also systems provide contextual information as a controllable and transparent resource, starting point, or option for users to amplify their action. The ideas were further elaborated by Chalmers and Galani (Chalmers and Galani 2004), who introduced the notion of seamfulness, where interesting aspects of context (such as caller ID) are made explicit for human interpretation, as opposed to seamlessness, the promise of automatic systems, which inevitably fail to deliver in many circumstances.
Recently and since the initial social science critique of context aware computing, much of the technical research has tended toward the enabling middleware layer for context aware computing, establishing ways of dealing with complex context, sensor fusion, and inferring logic for context recognition. However, since many of the components needed for designing and building context-aware applications and services are now available in mobile phones (integrated GPS, accelerometers), it is worth revisiting the design of context-aware services and applications to support human activity.

A CASE STUDY OF CONTEXT IN AGILE RIDESHARING

Social networking technologies have become widespread in the past few years in order to support communities, connect people to their friends, and expand the methods of social interaction on the web. The significant successes of emergent social technologies have brought new opportunities to enhance the usage and usefulness of context-aware applications. The new design challenge is how to devise ways to reconcile representational and social views of context. One of the proposed methods is to integrate and visualize physical values with data that emerges through interaction such as text messages.

We use a case study of agile ridesharing, the practice of ad hoc rides arranged in real time through mobile social software to elucidate context issues in the design of personal mobile social software applications. Currently, the prevalent focus of mobile business in travelling is on technical aspects such as algorithms for dynamic matching (Steger-Vonmetz 2005). However, Mobile social software has the potential to contribute to solving problems of meeting potential sharers, coordination, and logistics, while addressing privacy and security concerns. Decisions to meet and ride are highly contextual, depending on the circumstances of each person involved.

Why an agile ridesharing system?

Agile ridesharing systems hold promise because people often change their travel plans due to unforeseen circumstances. For example, User A stated her travel plans on Monday as follows:

“Ride on Tuesday and Thursday and car on other days but will leave early with husband before peak hour”

However, she changed her travel time late on Tuesday night because her daughter asked for a lift. “Confirmed late on Tuesday night that daughter needed a lift to uni by 9am so decided to wait to leave at 8.30am instead of 7am (before peak hour) and drop her on the way.”

Prototype

A prototype was designed to operate using a common web browser, so that it could be accessed using any web-enabled phone, laptop, or desktop, thus maximising the number of people who could participate in sharing. The prototype had a very limited functionality in that it only allowed people to send ride messages and information about seeking and offering rides. It was possible either to enter informal ride messages or in defined fields for origin, destination, journey start time, and whether a person was seeking or offering a ride. A screenshot is shown in Figure 1.

![Screenshot of the rideshare prototype](image)

Its purpose was to collect and share messages from people in the moment of travel in order to facilitate sharing, and, from a research perspective, to understand what they communicated in those particular circumstances. In this paper, we use example messages submitted to the ridesharing system of (Brereton and Ghelawat 2010), in order to examine how people represented their own context and the possibilities for context representation in agile ridesharing.

CONTEXT AND INTERACTION IN AGILE RIDESHARING

Before proposing how to integrate representational and interactional approaches to context, we first detail how context evolves through interaction. We take Dourish’s (2004) observations and demonstrate with examples from ridesharing.

1. “Contextuality is a relational property that holds between objects or activities”. For example, deciding whether to walk or rideshare depends on a number of relations: who you are meeting, and when they are likely to arrive, what the distance is, the weather condition and where you will go next, your mood or whether you car is clean.

2. “The scope of contextual features is defined dynamically, rather than being something that can be delineated and
defined in advance”. People usually sent simple messages to open a conversation “Walk to riverside anyone?” rather than specify details in advance. Often people preferred not to reveal their location to a variety of people for a variety of reasons (potential for robbery; surveillance; desire for privacy or mystery). However the example below indicates a particular context where GPS location became useful in the moment, even though it was not something either party typically wanted to communicate to the other. Two people confirmed a ride with this message:

“Leaving now from riverside. I’ll pick you up about 8:20 in green hill. Wait where I can pull over.”

The participants reported that once the driver hit traffic it became clear that he would be at least ten minutes late. The driver wanted to be able to communicate their whereabouts while driving. Meanwhile, the rider could not relax and read her newspaper, because she had to keep looking out for the car in busy traffic to be able to hop in at a moment’s notice. In addition, the driver did not know where exactly to pull over so they had to keep a careful eye out for the rider. On this occasion, the current location of both rider and driver came into focus as being an important contextual property to communicate.

3. “Context is an occasioned property, relevant to particular settings, particular instances of action, and particular parties to that action”. One participant messaged:

“City to dunmore. 4 hr parking limit. I'll be leaving at 11:30am for dunmore. Work in coffee shop til pic kids up in dunmore at 3. Then home to riverside. Anyone need meeting in cafe or ride to dunmore ?”

Constraints on the city parking and the commitments of this particular day would be hard to represent a priori or completely. Hence participants usually chose free form messaging over formal fields when wanting to represent complex circumstances of a particular occasion.

4. “Context arises from the activity”. Context is not just “there,” but is actively produced, maintained and enacted in the course of the activity at hand. For example a message conversation about having a coffee might go as follows.

“A: Want to go for coffee in uni café? 
B: I have a meeting right now.
A: Later then?
B: If we could meet in the city?”

We can see that the notion of a coffee in the afternoon in the city arises out of the negotiation. It could not have been easily computed a priori. Indeed significant a priori representation is often impractical for applications that support human activity.

**DYNAMIC CONTEXT CONTROL**

Brown and Randall’s strategies of allowing users to make sense of context information themselves and of using context information defensively (Brown and Randell 2004) still hold true. With more mobility and sharing of information, there is more and more context information that can be shared and the timeliness, security and privacy of information are increasingly important. For example, a simple dot on a map might represent the location of a mobile device. However, the dot does not indicate whether the mobile device is with its owner left somewhere, whether the system has recently updated the device location, or whether the phone/person/dot will be there for a while or about to leave. Other sensor information such as from the accelerometer in the phone might provide clues about when the device was last handled. Accuracy of the information can thus be increased, through utilizing more information from sensors, and through people annotating their status themselves. However, the important factor is that the device owner is likely to only want to share this information with particular trusted people at particular times when it is relevant and personally helpful to share such information. One of the interaction challenges thus becomes allowing people to easily select, in the moment, when certain relevant context features such as GPS sharing are available to others.

It is notable to contrast how little information people often need to state in text messages such as “pick you up from your house at 10” because friends know where each other are living. Often as little information is given as possible and it is given on a need to know basis.

![Figure 2. Using symbols to create representational context for the system in social messages and visualizing them.](image)

Thus the design challenge for mobile context is the artful combination of effective visualisation of sensor information, user provided content (text, speech or photos), and easily selecting when and with whom you share what. The traditional notion of context-aware research of simply gathering and integrating sensor data and providing it to applications needs to be revised as this approach can no longer be researched alone. Approaches need to consider

216
the provision of sensor data, its visualization, user annotation and user control all at once in the interface.

**Context Creation**

One approach to helping mine context information from free form messages in social platforms is the “hashtag approach” of Twitter in which users tag terms with a # sign in their free form messages to highlight to a system that these terms have a special meaning. In ridesharing, inserting the # sign before locations is one of the ways users could indicate to a system that a word indicates a location (Figure 2). Such approaches can help users to structure information in their free form messages and make them a computable component for the system.

Simple text mining can be used to suggest a few common conditions according to the keywords found in the message. (See Figure 3) The user then decides whether these conditions are appropriate for the inserted ride (Figure 3).

This approach aims to enhance the functionality of the system by showing better results for user intent or providing useful historical context for future activities.

Figure 3. Simple text mining of messages to suggest specific conditions in order to easily gather data about user intent

**Using historical context**

Even though, context is dynamic and emergent, past activity is useful to inform decisions. Applications use historical context in order to predict the future actions or intentions (Dey 2001). Knowing how many rides were shared between locations on previous days may be a helpful predictor of potential rides today. Since past activity has already occurred, it is stable, and yet it provides information upon which people can usefully draw to inform decision making for the present. Using data mining methods in order to summarize historical context for ridesharing sheds light on how many rides have been offered recently and between which locations, which can give an overview of system activity and ride potential without compromising privacy.

**Broader context**

Our rideshare trial indicated that people were interested not only in sharing rides, but also in sharing walks and public transport rides. Indeed, travelling together was seen as an opportunity to meet, and people used the prototype to organise meetings as well as rides. This result also indicates that the broader contextual information (about neighbourhood events or public transport) is important, both in being used for ridesharing and in representing the opportunity to rideshare in other forums, such as the local community digital notice boards, which include information about local events. The notion of establishing broader context is thus productive.

**CONCLUSION**

This paper takes a broader approach to context in design by considering the social and interactional issues of context-awareness along with the technical issues. Examples from agile ridesharing have been used to demonstrate that representational and interactional views of context are effectively reconciled by users in practice in the use of mobile devices. The contribution of this paper is demonstrating that interactional and representational views of context are not two distinct views. In particular this paper shows how to effectively reconcile representational and interactional views of context in design by moving beyond considering context awareness as a problem in isolation. This paper demonstrated how to take integrative approaches that carefully consider good visual design, user content, user control, historical and broader context information together in order to design a successful mobile context-aware applications.

**REFERENCES**


