PROXIMITY-BASED SYSTEMS: INCORPORATING MOBILITY AND SCALABILITY THROUGH PROXIMITY SENSING

A thesis submitted for the degree of Doctor of Philosophy

by

Caroline Sumathi Umashangar

Department of Information Systems and Computing,

Brunel University

March 2009

ABSTRACT

This thesis argues that the concept of spatial proximity offers a viable and practical option for the development of context-aware systems for highly mobile and dynamic environments. Such systems would overcome the shortcomings experienced by today's location-based and infrastructure dependent systems whose ability to deliver context-awareness is prescribed by their infrastructure. The proposed architecture will also allow for scalable interaction as against the single level of interaction in existing systems which limits services to a particular sized area.

The thesis examines the concept of spatial proximity and demonstrates how this concept can be exploited to take advantage of technological convergence to offer mobility and scalability to systems. It discusses the design of a proximity-based system that can deliver scalable context-aware services in highly mobile and dynamic environments. It explores the practical application of this novel design in a proximity-sensitive messaging application by creating a proof-of-concept prototype. The proof-of-concept prototype is used to evaluate the design as well as to elicit user views and expectations about a proximity-based approach. Together these provide a valuable insight into the applicability of the proximity-based approach for designing context-aware systems.

The design and development work discussed in the thesis presents a Proximity-Sensitive System Architecture that can be adapted for a variety of proximity-sensitive services. This is illustrated by means of examples, including a variety of context-aware messaging applications. The thesis also raises issues for information delivery, resource sharing, and human-computer interaction.

While the technological solution (proximity-based messaging) offered is only one among several that can be developed using this architecture, it offers the opportunity to stimulate ideas in the relatively new field of proximity and technological convergence research, and contributes to a better understanding of their potential role in offering context-aware services.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A-GPS	Assisted GPS
API	Application Programming Interfaces
CLR	Common Language Runtime
EDGE	Enhanced Data rates for GSM Evolution
E-OTD	Enhanced Observed Time Difference
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HCI	Human Computer Interaction
http	HyperText Transfer Protocol
IP	Internet Protocol
IR	InfraRed
J2ME	Java 2 Micro Edition
JVM	Java Virtual Machine
LBS	Location Based Service
LBT	Location Based Technology
MAC	Medium Access Control
NGR	National Grid Reference
NMEA	National Marine Electronic Association
PDA	Personal Digital Assistant
RF	Radio Frequency
RFID	Radio Frequency Identity
SMS	Short Message Service
TA	Timing Advance
UIN	Unique Identification Number
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Locators
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network

GLOSSARY

Some terminologies used throughout the thesis are specific to this research and therefore they are defined below. They will appear in the thesis with initial letter capitalised.

Clients: One of the two main elements in the Proximity-Sensitive System Architecture. This is a mobile device with various sensors and software application to support context gathering.

Context-Aware Service Architecture (Service Architecture): Architecture proposed for demonstrating how the Proximity-Sensitive System can be utilised to support proximity-sensitive services.

Environmental Sensors: One of the two main elements of the Proximity-Sensitive System Architecture and this is made up of exiting infrastructures and *ad hoc* sensor networks.

Explorer: A Software component in the Client (Proximity-Sensitive System Architecture) which is responsible for gathering context information about a mobile device's environment.

Integrated Sensors: Various sensors embedded or attached to Clients for context gathering.

Linker: A Software component in the Client (Proximity-Sensitive System Architecture) which is designed to provide the connectivity between Proximity-Sensitive System and other context-aware system components.

Messenger: Interface component designed for demonstrating the proximity-sensitive system.

Proximity-Sensitive System Architecture: An architecture proposed for supporting proximity-sensitive services. This includes Environmental Sensors and Clients.

Router: Designed to route messages between senders and recipients when demonstrating the proximity-sensitive system.

ACKNOWLEDGEMENTS

My effort in completing the PhD has been a challenging task and it would not have been a successful one if not for the tremendous support extended by the many that guided and helped me. I would like to express my gratitude to all those who have assisted me.

First and foremost, I would like to express my sincere gratitude to my first supervisor Dr. Mark Perry for his invaluable guidance and support. He gave me the freedom to work wherever and whenever possible and provided valuable advice and encouragement throughout my PhD. My special thanks go to my second supervisor Dr. George Ghinea for his support and constructive comments during the course of my PhD.

I would like to thank my fellow PhD student Steven Sams for helping me with server-side code. I appreciate his effort, expertise and his patience. I would also like to extend sincere thanks to my cousin and friend Ranjan for being there to answer my queries and doubts.

Well, within my kith and kin I received tremendous encouragement and support from all, especially from my aunty Leila who voluntarily agreed to proof read my thesis. I am grateful to my parents for their love, unconditional support and encouragement. I would not be where I am now without their guidance.

I have been fortunate to have a loving family. I am grateful to my husband Shangar for his endless love, immense support and encouragement throughout. It would have been impossible to carry on and complete my thesis without his friendship and understanding. Finally, I am thankful to my wonderful sons Angelo and Stephon for the understanding they have shown and inspiration they have given me.

Chapter 1: Introduction

1.1 Overview

The thesis examines the notion of spatial proximity and technical approaches to support proximity sensing as means of offering context-aware services. Alongside this, the thesis seeks to broaden the knowledge necessary for developing such services on mobile devices. The majority of existing context-aware systems are location-based and infrastructure dependent. As a result, these systems are limited in their ability to operate outside the area covered by their system infrastructure or to support context-aware services in highly mobile and dynamic environments where users' mobile devices, and the entities with which they interact, may all be mobile. In addition, current systems have focused on single level interaction, thus interactions cannot be scaled to different levels of spatial granularity.

As a step towards addressing the above limitations, the thesis discusses the design of a Proximity-Sensitive System Architecture based on the concept of proximity and technological convergence. It explains how this architecture interconnects and interacts with the other components of the Service Architecture to offer proximity-sensitive services to mobile device users. The thesis details how the proposed architecture is developed into a proof-of-concept prototype. Alongside the development work, the thesis critically explores various development platforms, programming languages and tools for this type of prototyping, and draws out the experiences and lessons learnt during the development process. Finally, the thesis explains how the proof-of-concept prototype (also referred as proof-of-concept) is used to qualitatively evaluate the design, and the nature and role that spatial proximity might play in providing communication services.

1.2 Research Background

Today's mobile communication systems support voice, text and multimedia messaging, allowing mobile device users to create and send messages to other individual mobile users or a group of mobile users. Such messages are delivered instantaneously to the connected recipients regardless of their current context. There are often situations where these types of 'anytime anywhere' systems provide users with information that is irrelevant to their current context resulting in information overload or out of context information delivery (Perry *et al.*, 2001).

Efforts to overcome this problem led to the development of several systems that support context-aware messaging services (e.g. Marmasse, 1999). The majority of these are location-based (Schmidt *et al.,* 1998; Dey and Abowd, 2000) and application specific, and rely heavily on the underlying static infrastructure (Nelson, 1998; Mitchell, 2002). In such systems all their infrastructural elements are static. Thus, these systems still have 'fixedness' in them, allowing mobile devices to communicate only within a particular environment through static elements of their infrastructures, contributing very little to our understanding of the wider potential of the underlying mobile and wireless technologies (Coulouris *et al*., 2005). This 'fixedness' also makes such systems less flexible to adapt to changes in the environment (indoor and outdoor) and unsuitable for wide-scale adaptation (Hong *et al*., 2007; Riva and Toivonen, 2007).

More importantly, a location-based approach to context-awareness overlooks many interesting aspects of mobile communication, such as mobility and scalable interactions which could be exploited to provide more selective and targeted messaging services. The absence of a common and generic architecture to support these important features, has led this research to take a very different approach and to focus on spatial proximity where context-awareness is driven by proximal relationships between connected entities, giving all such entities the freedom to move around if necessary and still take part in supporting context-aware services.

The thesis describes how the above limitations can be addressed through the design of a Proximity-Sensitive System Architecture. It raises several questions relating to the design, and the type of technologies that can be used to support proximal relationships: How can these technologies be used to support services outside the areas covered by infrastructures? How can they help to achieve maximum coverage? How do they offer mobility, particularly when all entities involved are mobile? How can they support scalable interactions? To find answers to these questions, the thesis begins with an exploration of the unique characteristics (e.g. coverage range, physical features installation and maintenance requirements) of various sensing technologies. This exploration has led to the understanding that there is no single technology that is capable of offering all of the characteristics that are required, most notably, ubiquitous coverage, mobility and scalability (where scalability is defined as the ability to offer communication at different levels of spatial granularity). Based on this conclusion, a decision was made to take advantage of what has been termed 'technological convergence' to bring together different types of sensing technologies with their distinctive characteristics into one unified system, and to develop them to work in a complementary manner to achieve all of the required features.

The convergence of mobile devices and wireless technology helps mobile devices to sense their surrounding environments through their embedded sensors. The design described in this thesis not only utilises convergence of mobile and wireless sensing technology, but also takes advantage of various wireless sensing technologies and brings them together into a single mobile device. The following sections describe and define some of the terminologies used, and provide a brief background for the work described in this thesis.

1.2.1 Context, Context-awareness and Context Types

The literature contains several examples of attempts to define context within different domains but of special interest for this thesis, are the definitions used within context-aware systems and services. Further, the focus is on the information implicitly gathered by sensors on the device, rather than that explicitly provided by users.

Dey (2000) defines context as any information that can be used to characterize the situation of an entity. An entity is a person, place or an object that is considered relevant to the interaction between a user and an application, and it includes both the user and the application itself. Dey also draws attention to the lack of range in terms of the context types used to offer context-aware services, and the fact that most systems have focused on location. E-graffiti (Burrell and Gay, 2002), comMotion (Schmandt *et al*., 2000) and Siemen's 'Digital Graffito' (Weber, 2005) are examples of such location-based systems.

The thesis argues that the context type 'location' is static, whether presented as geographical coordinates, places or static entities (such as buildings, doors, walls, floors and desktop computers), and provides services in relation to those locations. Yet, for many forms of context-aware applications, information beyond location is required. Below are two examples of such situations that cannot be supported by location identification alone. The first scenario explores a situation where messaging takes place when two mobile entities coexist. The second explores a situation where messaging takes place when a mobile entity coexists with another entity (not necessarily mobile), emphasising the importance of proximal range between the entities involved in the messaging.

1.2.2 Issues with Location-Based Systems

This section discusses two theoretical messaging scenarios and tries to explain why they cannot be supported by current location-based systems. Most current location-based systems rely heavily on static network infrastructures, referred as infrastructure in the rest of the thesis. These static elements help to sense (discover) mobile devices using location-inferred discovery: the location of the infrastructure and its elements are already known, so the system deduces that mobile devices discovered by these elements are also close to this particular location. However, this type of discovery (i.e. location identification) is not always sufficient to provide context-aware services, nor will building an infrastructure to provide coverage in a room or a building be of much help in providing services outside this area. In addition, some services may need to be delivered in relation to a context condition other than location (e.g. proximity). To discuss this in more detail, the example scenarios are examined below.

Scenario 1: Infrastructure and Location Independence

A typical example for the first scenario: Andy and Ben are having a meeting. Ben is due to meet Cathy in the near future and Andy needs to send a message to Cathy about Ben. The message could be a simple reminder between friends, some medical notes, employee details or business details, but the details contained in Andy's message are only relevant when Ben is with Cathy. Ben's meeting with Andy, and then with Cathy, could take place anywhere; it is not necessarily going to take place in a previously known location, place or at a set time. It may even be an *ad hoc*, serendipitous encounter. Further, the subsequent meeting between Cathy and Ben is likely to take place in a different location to Andy's and Ben's original meeting place, and may occur in settings as varied as an outdoor environment, inside a work building, in a bar, or even whilst mobile in a car or train. This situation is illustrated in Figure 1.1, where the initial meeting between Andy and Ben takes place in Andy's office in London, and the subsequent meeting between Ben and Cathy takes place in a Café near Cathy's office. Cathy may not be working in the same London office, so the Café could be anywhere, and Andy may not have visited it before. Further, the meeting may have been arranged at the last minute (i.e. decided to meet in the café). The message delivery in this instance is not related to the location of the user. In addition, there is uncertainty regarding the meeting venue. Under these circumstances, the messaging cannot be supported by location-based systems such as comMotion and Floating Note (Multaharju, 2004).

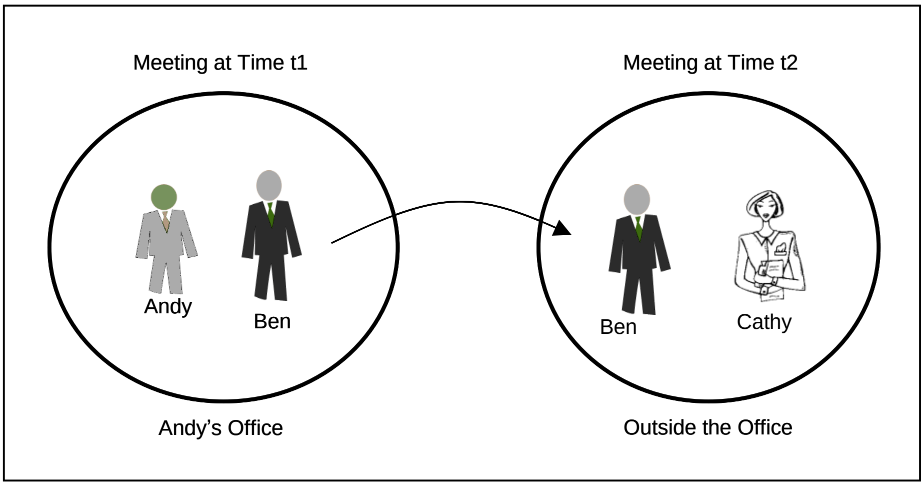

Figure 1.1: Infrastructure and Location Independence

The scenario described above is recognisably commonplace in many people's lives, and yet, in spite of the popularity and ubiquity of mobile and wireless technologies, a technological solution that addresses this need is currently not available. What is particularly interesting about this scenario in terms of system design is that techniques of location-inferred discovery are not sufficient on their own for this type of messaging, as these would require advance information on *where* the second meeting was going to take place. One way of overcoming this limitation would be to move away from location and focus on the spatial relationship between entities.

Scenario 2: Spatial Granularity of Proximal Messaging

The second scenario highlights the issue of spatial granularity. A sender wanting to leave a message for someone may well want the message to be delivered only when the recipient is within close range of the message's target but, problematically, the notion of 'close' is one that has a wide range interpretation depending on the nature of the message.

Consider a situation where Andy wishes to leave a message for Ben. Andy's message is only relevant when Ben is in the vicinity of a particular object. This object could be fixed or mobile, for example, a housing estate, a building, an office door, a fridge, a key or a book. Andy needs to see Ben to get a document signed off by him before meeting his clients. He goes to see Ben, but Ben is not at his desk. Andy decides to leave the document in Ben's in-tray. In addition, he wants to leave a message telling Ben what needs to be done, to review and return the document to him before his meeting. Andy wants this message to be delivered only when Ben is back at his desk, not while he is some metres away having a conversation with another person, say at that person's desk. Figure 1.2 shows (left) Andy coming to look for Ben whilst he is away from his desk, and (right) Ben coming back to his desk.

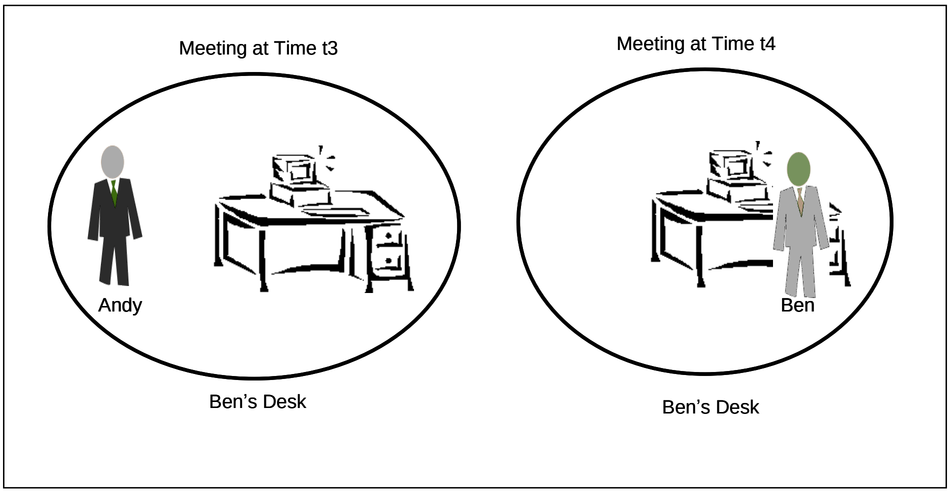

Figure 1.2: Granularity of Proximal Messaging

As with the previous scenario, the need for messaging at these different levels of proximity is again recognisable as extremely common. Current non-electronic techniques to deal with these issues, without using technological solutions, include the use of post-it notes, blue-tacked sheets or paper, or even graffiti, as appropriate to the zone of coverage required. However, these media are not particularly interactive (i.e. a two way dialogue is hard to establish), nor do they offer much in the way of media appropriate to the setting (for example, paper supports textual and graphic images, but does not support audio or photographic representations). This type of messaging between people covers very different 'zones' of interaction (e.g. in relation to a building, an office door, a book), and so specifying a generic messaging proximity in such instances (such as 'within 10 metres') may be inappropriate.

In order to support the kind of messaging scenarios discussed above, this thesis explores the concept of spatial proximity and presents the design of a system architecture that is based on this concept. It critically examines the design issues involved in supporting ubiquitous coverage (in terms of different types of environments, entities and technologies), mobility and scalability (of the proximal relationships). The next section defines spatial proximity and describes its unique characteristics in terms of their relevance to the work discussed in this thesis.

1.2.3 The Concept of Spatial Proximity

The integration of mobile and wireless sensing technologies, and the widespread penetration of wireless sensors in our environments have not only given opportunities to offer a wide range of services but also have created new challenges in terms of system design. Location as a static form of context is no longer adequate to support the mobility and dynamic changes introduced by mobile and wireless technologies i.e. location always supports communication in relation to a static sensor or geographical coordinate and fails to support communication where all entities involved are mobile. This thesis investigates a different approach, namely a type of context called *spatial proximity*, referred to hereafter as proximity.

Proximity is defined here as a spatial relationship between entities, i.e., the perception of being close to an entity, a person, place or an object. Entities such as people and some objects are not always static. This is very different to location-based systems which while taking account of moving device users or objects that need to be located (such as freight), have not considered moving entities in their surrounding environments. Location-based systems discover mobile device users or mobile objects in geographical coordinates through GPS (Global Positioning System) (Cheverst *et al.*, 2000) and location databases (LaMarca *et al.*, 2005), or in relation to static entities (Want *et al.*, 1997).

In contrast to location as a concept for context-awareness, proximity provides support for discovery through spatial relationships. These spatial relationships rely on spatial coexistence, where coexistence is defined as co-present entities in an environment that are *close enough* to be sensed (i.e., discovered) by each other. Such spatial relationships can occur between a mobile device and a static entity (e.g. station, desktop computer) or a mobile device and a mobile entity (e.g. mobile phone, book, laptop computer), thus offering support for context-aware services in relation to both mobile and static entities. However, proximity should not be confused with relative location which only supports a relationship between an entity and a static entity, where the static entity resides in a location.

Proximity has another interesting characteristic: a scalable spatial relationship, a relational association naturally derived from spatial distance. Being close to something can be interpreted in many different ways, for example, near the station, very near the station or almost at the station. This relational association makes it possible to use the spatial distance between entities to offer communication at different levels of spatial granularity. This thesis explores the possibility of enabling this use of scalability through a proximity sensitive architecture that takes advantage of the technological characteristics enabled by multiple sensing technologies.

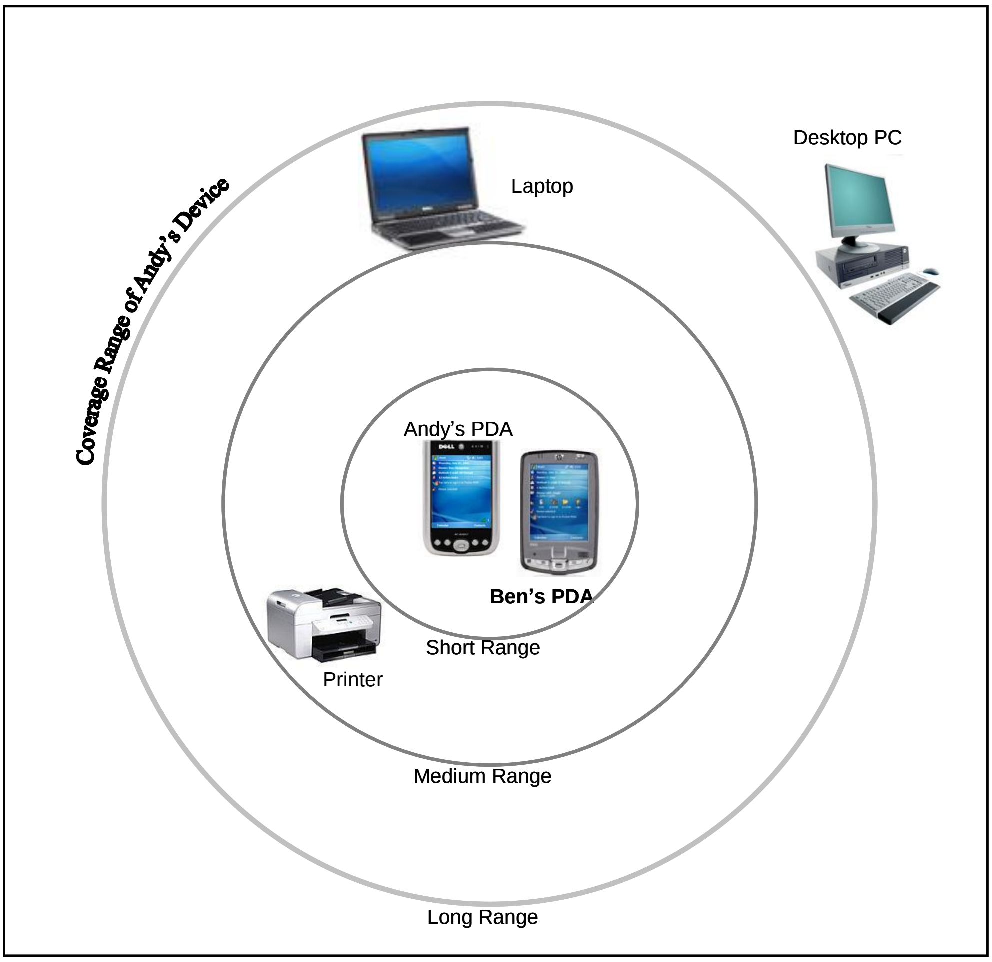

Figure 1.3: Proximity Sensing

Figure 1.3 illustrates how a mobile device (Andy's PDA – Personal Digital Assistant) can discover other coexisting mobile (Ben's PDA and laptop) and static (printer) devices at different levels of granularity. For example, Ben's PDA and Andy's PDA are almost in the same place (short range), the printer is 'very close' to Andy's PDA (medium range), the laptop is 'close' to Andy's PDA (long range) and the desktop PC is outside the discoverable distance of Andy's device. This scalability can be used to provide interaction at different levels of scale, Andy's PDA or mobile phone is on his office desk, in his office or somewhere in the office building.

1.3 Motivation for the Research

The work discussed in this thesis is motivated by the identified lack of support for mobility and scalability within existing context-aware systems. Alongside this, is the interest shown repeatedly by conferences such as MobiQuitous (from 2004 to 2008) in solutions that support communication services where all of the entities involved are mobile. Furthermore, the conference organisers have highlighted that designing such systems pose many challenges as they need to move beyond ‘fixedness’ to provide services everywhere. This gave the motivation to explore the possibilities of providing mobility to communication systems, more precisely to context-aware systems.

From an academic point of view, this research aims to examine the possibility of enabling context-aware services across different environmental settings (indoor, outdoor, and outside areas covered by infrastructures), in relation to people, place and objects, and at various levels of spatial granularities. Technical issues relating to system development are also important to this research as the design has to be developed on a mobile platform. Mobile application development, especially where it involves convergence, is a fairly new area of research that presents compelling challenges to developers and would benefit from further research. The thesis therefore reflects on the problems faced and the lessons learnt during the creation of the proof-of-concept prototype. In addition, the thesis develops a proof-of-concept prototype which helps better understand users’ views on, and their vision for such systems. This user study provides a resource for future academic researchers and designers, as well as offering some valuable information for commercial mobile application developers and service providers, such as mobile network operators.

1.4 Aim and Objectives

The main aim of the thesis is to examine the possibility of providing mobility and scalability to context-aware systems through the use of a proximity based approach. Its secondary aim is to further the understanding of mobile application development that involves technological convergence.

In order to achieve these, the following objectives were set:

1. **Identify challenges faced in designing context-aware systems**
   Review previous research into context-aware systems and critique their approaches in adapting to mobility and scalability. To understand the problem in detail, draw attention to their design constraints and elicit the kind of characteristics necessary for supporting mobility and scalability.

2. **Examine proximity and technical approaches to proximity sensing**
   Explore the potential of proximity and how it can be used to offer different set of features compared to those of location. To do this, examine the concept of proximity, provide definition for proximity, explain and interpret its distinctive characteristics and explore how it can be used to address the design issues faced by current context-aware systems. Identify technical approaches to support proximity sensing.

3. **Design and critically examine the Proximity-Sensitive System Architecture**
   Create a Proximity-Sensitive System Architecture based on proximity. Understand how this architecture fits into the broader area of context-aware services and how it takes advantage of spatial relationships and technological convergence to obtain contextual information and interpretation necessary for addressing the above issues.

4. **Investigate the practical issues relating to mobile application development**
   Examine the practical implications of this architecture by creating a proof-of-concept for proximity-sensitive personal messaging service, and how they can be implemented using existing development environments and tools. In addition, identify issues relating to the implementation of mobile applications where it involves technological convergence.

5. **Evaluate the design approach and architecture**
   Use the proof-of-concept as a technology probe to evaluate the proximity based approach by conducting a user study, and provide a reflective review of the Proximity-Sensitive System Architecture, discussing its merits and limitations with regards to its extendibility, adaptability, coverage, mobility, scalability, reliability, applicability and privacy.

In summary, this thesis shows how scalable context-aware interactions can be supported in highly mobile and dynamic environments by taking advantage of the unique characteristics of proximity and technological convergence, in particular the integration of multiple and diverse sensors on mobile devices. By describing the creation of a proof-of-concept Proximity-Sensitive System, the research enables a greater understanding of the design considerations and issues surrounding context-aware systems. Moreover, the creation of a proof-of-concept contributes to a technological tool to evaluate the choice of approach for supporting mobility and scalability within context-aware systems. The next section discusses the research method that helps to achieve the aims and objectives of the research.

1.5 Research Method

In this thesis the research aim is achieved through the development of a proof-of-concept for a Proximity-Sensitive Messaging System. This proof-of-concept is developed as a tool for learning and communicating, rather than for testing and quantitatively evaluating its technical optimisation or commercial applicability. This means that the primary focus is not on the proof-of-concept alone, but also on the process of developing the proof-of-concept.

The research method for developing the proof-of-concept consists of the three steps that are standard practice in technical research of this nature: Analysis, Design and Implementation. The Analysis will help derive a set of requirements by means of a reflective critique of related systems and through an analysis of the system's desired characteristics. This will in turn help identify the gaps in the current literature and contribute to a better understanding of the context-aware system that is to be developed.

In the design phase, the requirements identified in the Analysis phase will be used to build a system architecture by a process of examining various design options and selecting the one that best meets the research requirements. The decisions made and the reasoning behind them**,** and why a particular option was selected and others rejected**,** will be also be described to broaden the understanding of this design space and be useful to future researchers interested in finding out what alternatives had already been considered and why they were rejected (Burge and Brown, 2000).

Finally, in the implementation phase, the design will be transformed into a proof-of-concept, which will be used as a tool for evaluating the proximity based approach to service delivery. This last phase will help identify the practical issues relating to the implementation of such systems on mobile devices, which it is acknowledged is not a straightforward process. The proof-of-concept created will then be used as a technological tool to encourage reflection and discussion, and to answer and identify questions relating to and arising from the approach. In the absence of a tool it is rather difficult to communicate novel ideas and concepts to the non-technical and non-research community as there is very little understanding about the system or service that is to be created, and the proof-of-concept offers a physical artefact to inspect and critique in this respect:

> *Getting feedback of potential users and ultimately buyers at an early stage is very beneficial for the development of technologies. If there is no communication with potential users there is a serious risk that research will explore issues that are of no interest to anyone. On the other side, potential users will often not consider their needs and requirements because the technology is very abstract and rather recite ideas from the science fiction genre.* (Schmidt, 2002:217)

For this reason, the external validity of the system is considered and examined through the usefulness of the proposed system to potential users. To be considered as useful, any mobile communication service should provide users with ease of use, ease of adoption, efficiency and cost effectiveness (Kaasinen, 2005). Furthermore, it should assure them the level of privacy they desire (Neisse *et al.*, 2006). Another consideration is that the requirements of individuals who may use mobile communications for work or social interaction are different from those of professionals whose communication requirements may be more demanding and specific to their professions. Thus to be successful, designers and service providers will need to understand and potentially cater to these differing needs.

One way of ascertaining what these needs are is to ask existing mobile communication system users and potential users themselves. Thus a decision was made to interview a sample of potential users on (1) their understanding of the concept of proximity (2) their specific requirements with regard to social as distinct from professional requirements (3) what their expectations were with any new proximity-based systems or services that may be offered in the future and (4) other general concerns they may have.

The data requirements dictate the type of sample that is to be used in the study. In this instance, to provide any meaningful input, the sample had to have some practical experience with mobile communications, at least some of the sample should be able to express their specific professional needs, and at least some their social interaction requirements of mobile communication. They need to be able to formulate their needs in a way that could be amenable to a practical solution through new services or by enhancement of existing services. They also should be able to think beyond their immediate application needs to other considerations such as practicality, efficiency, and personal issues that are relevant.

Given these specific requirements, and constraints of time and finance that rule out a large-scale study, it was decided that an appropriate sample would be a purposive sample (Trochim, 2006), small in number, and with the interviewees being selected from different professions and age groups. One common feature would be that each member of the sample group would have practical experience of mobile communications either in a personal context or a professional context or both. The sample will be interviewed using a set of questions as a guide to trigger open-ended discussion in a face-to-face situation to allow full expression of individual views and thoughts on their needs and limitations of existing services, as well as other issues which they considered important. The design of the study and how it was implemented, the findings and relevance for future design and service provision are described in Chapter 6.

1.6 Thesis Scope

The scope of the work described in this thesis covers the architectural design of proximity based system. Its intention is to highlight that proximity offers very different characteristics compared to location, and to demonstrate how the concept of proximity can be exploited to offer different services. Alongside this, it aims to understand the practical challenges in implementing such systems on resource constrained mobile devices. However, this thesis does not intend to provide solutions for useable interface designs or optimum connectivity for proximity sensitive services.

1.7 Contribution Summary

The thesis focuses on a system architecture for providing mobility and scalability to systems through proximity sensing. It does this through existing infrastructure and a variety of current ad-hoc sensor networks. Although it discusses a technical solution to proximity-sensing and describes its use in various context-aware personal messaging scenarios, its use goes beyond the sensing technologies discussed in this thesis. The Proximity-Sensitive System Architecture is designed to adapt to any current or future sensing technology that has a unique identifier or a mechanism for providing location information.

Additionally, the Proximity-Sensitive System Architecture can support a wide variety of applications outside context-aware personal messaging services. The user study identified Proximity-Sensitive System's use in context-aware information delivery (e.g. tagging of offenders, tagging medical notes to people, delivering marketing and local information). This architecture could also be extended to mobile gaming applications where a user's proximity to location or a mobile entity is required to trigger events.

The Proximity-Sensitive System design itself helps to stimulate ideas in relation to technical characteristics of various sensors and their benefits. The design process explains how technical limitation of sensing technologies can be used to system's advantage. For example, the reasons why a user would tag a space to a range of a few centimetres are likely to be different to why they would do so where the range and accuracy of the tag is in the order of a radius of 10, or 100 metres. Thus, sensors with few centimetres coverage range have the advantage of providing fine grained sensing to systems. The discussion on characteristic variation among sensors helps to illustrate the resultant implications for messaging: most have different communicative affordances, and offer different opportunities for communication. Clearly, the particular constraints and properties of sensors used is likely to have a major impact on the ways that they are incorporated in user practices, and how they can be better used to meet service requirements. In addition, bringing together multiple but different types of sensors into Proximity-Sensitive System Architecture helps to demonstrate that applications built on top of it can be used to provide a variety of services, due to the different capabilities and constraints that they carry.

1.8 Thesis Layout

Chapter 2 – Analyses and compares technologies, platforms and tools to provide the background necessary for the work discussed in this thesis. In addition, it presents a literature review on relevant systems and prototypes; explains and critiques their approaches to determine the main issues that need to be addressed to support context-aware services in mobile and dynamic environments.

Chapter 3 – Discusses the main issues that are expected to be addressed and the motivation behind them. It introduces innovative ideas to address coverage, mobility and scalability. It describes how these innovative ideas evolve into a Proximity-Sensitive System Architecture. In addition, it explains how Bluetooth, RFID and GPS technologies can work in concert with one another to gather contextual information. It also provides a rationale for the approach adopted, the context type and technologies chosen to address the main issues discussed in this thesis.

Chapter 4 – Examines the role of Proximity-Sensitive System Architecture in a Context-Aware Service Architecture (referred to hereafter as Service Architecture). It deals with the design of a Service Architecture for supporting messaging applications and describes how the proposed Proximity-Sensitive System Architecture fits into the overall Service Architecture i.e. that the Proximity-Sensitive System Architecture is a subset of Service Architecture and is responsible for providing the contextual information necessary for supporting context-aware services.

Chapter 5 – Covers the platforms and tools used in prototyping, and the rationale for particular choices made. To examine the architectural and technological features of the proposed Proximity-Sensitive System Architecture, the chapter describes the creation of a proof-of-concept Service Architecture. The challenges faced in various stages of prototyping are also addressed, and recommendations are discussed.

Chapter 6 – Evaluates proximity based approach and Proximity-Sensitive System Architecture through a user study. In addition, it provides a reflective review of the design in terms of ubiquitous coverage, adaptability, mobility, scalability (scaling interaction to different spatial granularity), complexity, extendibility (to larger area), security, privacy and reliability.

Chapter 7 – Provides a summary of the thesis. The conclusion drawn from the research and contributions made to address the current issues are discussed. In addition, extensions and suggestions for future work are indicated.

Chapter 2: A Critique of Technologies and Systems for Context-Awareness

2.1 Introduction

The focus of this thesis is to examine the possibility of introducing mobility and scalability into context-aware systems through proximity sensing. The Chapter lays the foundation for this by examining the terminologies and technologies that are relevant to context-aware systems. The current approaches for designing context-aware systems are also discussed, highlighting knowledge gaps and identifying research needs that are to be addressed by this thesis.

2.2 Understanding Mobility

Interest in mobile technology has increased dramatically in recent years. This has introduced new challenges in terms of mobility, and how this characteristic can be incorporated into future systems to provide services that utilise mobile technology. The thesis focuses on mobility that is associated with some form of movement in space. Mitchell (2002) classifies mobility into two main types: user mobility, and user and device mobility. This thesis introduces a third type of mobility, user device and entity mobility to support the mobility that is introduced by new mobile and wireless technologies. Each of these mobility types is described below.

2.2.1 User Mobility

In user mobility, a user moves across a variety of environments and use computing devices to gain access to information. In this, users do not carry computing devices instead they move from place to place and use computers found in those places. User mobility focuses *only on the mobility of the user, who for example, moves around an office building using fixed computing devices.* (Mitchell, 2002:217)

2.2.2 Device Mobility

Device mobility is very different to user mobility. In this, users move around with their mobile devices. User mobility *involves the mobility of both user and computing device, for example, a field engineer using a pen-tablet and working on the move.* (Mitchell, 2002:217)

Device mobility allows users to interconnect and interact with information space or services through their devices (e.g. wearable computers, personal devices, integrated systems). Users could access internet services by connecting their devices to mobile phone infrastructure or WLAN access points wherever they are. Additionally, they can interact with other devices using built-in sensors embedded into their devices.

2.2.3 User Device and Entity Mobility

The combination of mobile and wireless technologies have contributed to a highly mobile and dynamic environment in which users, their devices and all entities involved can be mobile. An example scenario discussed in Chapter 1 already gives a realistic dimension to this type of mobility, a user (through his or her mobile device) may want to communicate with another mobile device user in relation to a third mobile entity (referred as third party device in this thesis). In this, the third mobile entity could be a mobile phone, PDA (Personal Digital Assistant) or any sensors enabled object that is likely to move from its current position. This type of mobility encompasses more than just the surrounding environment of geographical coordinates or static entities. In order to support services based on this mobility it is necessary to extend context-awareness beyond locations and areas covered by infrastructures.

In summary, to support context-aware solutions on mobile platforms, the thesis focuses on the mobility that involves device (mobile devices) movement. There is considerable evidence to suggest that mobile devices are an important part of every day life for many people on the move. It is stated in Abowd *et al* (2005) that such devices (e.g. Smartphones) are realistic platforms for everyday pervasive computing applications. Based on the literature findings the thesis focuses on mobile devices as a platform for providing context-awareness to users. As a result, user mobility is excluded in our discussion, and the device, device and entity mobility become the main focus of this thesis.

2.3 Understanding Scalability

People often wish to communicate at various levels of scale, from making a note on a small object (e.g. a book, a desk, a bag), objects, people or places at a larger range (e.g. a room, a person, a vehicle), or to post information over a wide area (e.g. a building, a shopping centre, car park). We as users may therefore wish to *attach* information onto objects that are both mobile and static, which can only meaningfully be interpreted when the message annotation is scaled to an appropriate distance from the entity that it is appended to. Scalability is introduced to allow users to delimit and determine the proximity range of a physical area for messaging. By limiting message delivery to a particular proximity range, users can post messages where they are expected to be most relevant. For the purpose of this thesis, the proximity range is grouped into short-range (a few centimetres), medium-range (up to 10 metres), and long-range (over metres). This will help to offer more targeted short range (e.g. leaving a note on the fridge), medium range (e.g. room level information delivery) and enhanced long range (e.g. traffic information) context-aware services to users. In addition, the long-range could be further divided into different levels (e.g. 100 metres, 500 metres, 1 km) to tailor long range delivery to different type of services.

2.4 Technologies for Context-Aware Systems

Advances in mobile and wireless technologies have made communication and information services available almost everywhere. This section examines these technologies with the hope of taking communication and information services beyond the availability and accessibility, making them adaptable to situations and environments in which they are used. It reviews technologies with the intention of augmenting mobile devices with awareness of their current environment, helping devices find who and what is present in the current situation, and tailor communication and information delivery accordingly. It looks for technologies with unique characteristics that will help to design scalable context-aware systems for mobile and dynamic environments. These technologies are grouped into mobile, wireless networks and sensing technologies based on their main functionality (i.e. solution for mobile platform, network connectivity or sensing) and discussed below. In addition, it examines development frameworks for implementing context-aware services on mobile devices.

2.4.1 Mobile Devices: A Platform for Context-Aware Services

Mobile devices give the freedom and flexibility to mobile users to communicate and work without being tied to desktop PC and fixed telephone lines. People these days move around and use different types of mobile devices to access information and communicate with others. These mobile devices come in different sizes, shapes and more importantly with different functionality. They are generally pocket size devices with limited processing power, battery life and small screens. Examples of such devices include wearable computers, mobile phones, palm tops, PDAs, etc. However, the research in this thesis focuses on providing services on devices that fit in with the use of existing and commonly used mobile devices. In this way, people will not need to add to their existing complement of devices that they carry around with them to access context-aware services. This section therefore limits the discussion to currently available (2004-2008) mobile phones and PDAs, and provides a brief overview on examples of these devices, their main functions and the wireless technologies that they support.

Mobile phone technology is progressing rapidly, with phones being introduced with Wireless Application Protocol (WAP), built-in cameras (Samsung D900), video recorders (Nokia 6500 classic), FM radios (Nokia 5200), wireless network connectivity, mp3 players (Sony Ericsson W880i), games, GPS (Nokia N95), RFID readers (Nokia 5140), Bluetooth (Nokia N60 series), WLAN (O2 XDA) etc. People are able to use their mobile phones to make voice calls on the move, send and receive text, multimedia messages, take photos, listen to music, play games and for browsing internet. However, to date they have not yet evolved to support windows applications such as word processing packages in similar fashion to PDAs.

The PDA is another type of mobile device that is becoming powerful enough to replace desktop PCs in offering data access, to-do lists, Day Planner, Excel, Word processing and many more applications. In addition, PDAs are now being equipped with variety of built-in sensing technologies (e.g. WLAN, Bluetooth and InfraRed (IR)). Some also have additional expansion ports for incorporating external hardware. However, the main issue that causes problems for PDAs is the wireless connectivity; they provide connection via WLAN networks and fail to support mobile phone networks. To overcome this, and combat issues faced by mobile phones, PDAs and mobile phones have been combined into a single mobile device. This new device is often referred as a Smartphone. Such

Smartphones combine functionality from both mobile phones and PDAs. Devices such as Blackberry (Blackberry, 2008), Treo (Palm, 2008b), and windows mobile based Smartphones have become an integral part of many mobile users' lives and it seems that more people are carrying these devices than ever before (Riva, 2007). As a result, these devices can be found around people in their homes, work places and in other public places.

Based on the availability of these Smartphone devices, it seems that they could offer a rich platform for offering context-aware services. Nevertheless, mobile technology alone is not sufficient to provide context-aware services, and devices need to be augmented with context-awareness, broadly, the knowledge of who and what is nearby. Then, the information relevant to that particular context must be delivered to and from mobile device through a wireless network connection. The candidate technologies for enabling network connectivity and context-awareness are discussed in the next two sections.

2.4.2 Wireless Networks: Technologies for Connectivity

Wireless networks connect devices to other devices and networks without physical ('wired') connections. The lack of a wired connection means that users are free to move around and still have access to remote information and other devices wherever they are. There are different types of technologies present in mobile devices to provide the connectivity necessary for communication. Wireless local Area Networks (WLAN) and mobile phone networks are two of the main technologies used by mobile devices to connect to other devices and remote resources. Infrastructures necessary for these technologies already exist in the environment, although WLAN is not yet fully pervasive to provide ubiquitous coverage.

The mobile phone networks were initially designed to support voice communication on mobile devices such as mobile phones. In recent years, this network has extended its services to data communications through Global System for Mobile communications (GSM) e.g. SMS, General Packet Radio Service (GPRS), Enhanced Data rates for GSM Evolution (EDGE) and Universal Mobile Telecommunication System (UMTS). In mobile phone networks, Base Stations are built to provide coverage for an area, and each Base Station is responsible for providing the connectivity to mobile phones within its area. The current mobile networks are designed to provide coverage even inside buildings and vehicles. This is currently a mature network that provides almost ubiquitous coverage, has the potential to offer network connectivity almost anywhere.

A WLAN is a wireless network which links two or more computers without wires. WLAN uses radio signals to enable communication between devices within a limited area, often referred as a basic service set. This gives users the mobility to move around within its coverage area and still be connected to the wireless network (Seppanen, 2002). In a WLAN network, all computers that can be connected to a wireless network are called stations, and they fall into two categories: access points and clients. Access points are base stations for the wireless network and they transmit and receive radio signals for enabling communication between WLAN enabled devices. The clients can be mobile devices such as laptops, PDAs, phones with Internet Protocol (IP), desktops or workstations that are equipped with WLAN interface. All the WLAN stations that can communicate with each other form a Basic Service Set (BSS). There are two types of BSS: Infrastructure BSS and Independent BSS (IBSS). In infrastructure BSS, access points help the devices in one WLAN network to communicate with other networks, and obtain information from remote servers. However, IBSS supports client based ad-hoc connection (peer-to-peer) allows wireless devices within range of each other to discover and communicate directly without involving central access points. WLANs are commonly found in mobile computing devices such as PDAs, Smartphones and laptop computers, and this provides flexible wireless connections for accessing information resources such as the internet and mail servers when working away from home and offices. Their coverage range varies from 10 metres to few hundred metres depending on the type of specification. Examples of WLAN specification types include 802.11a, 802.11b, 802.11n and 802.11g.

In addition to the above two wireless technologies, there are other technologies such as Bluetooth for enabling wireless connection between devices. These technologies provide access to information held on those devices. During the development stages of this research, Bluetooth technology was unable to provide direct access to external information servers; thus a Bluetooth enabled PDA cannot directly access internet, it requires a Bluetooth enabled phone or laptop with WLAN as a modem to get access to the internet and remote servers. Bluetooth has since evolved into a technology that provides mobile devices with easy, secure, inexpensive, and high-speed connectivity to the internet through its access points (msmobiles, 2005), although this idea was developed too recently for consideration in this thesis.

2.4.3 Sensors: Technologies for Context Gathering

Sensing technologies have made it possible to embed and relocate various sensors on people and objects, and within places (e.g. Hewlett-Packard's Cooltown) in the environment. At its simplest level, these sensors may be used to sense the presence of another device (for example, the Lovegety, see Iwatani, 1998) or its presence in relation to other devices or location, to support communication, data transfer and resource sharing (for example, printers and fax machines). To broaden the understanding on how these sensors can be utilised to offer mobility and scalability, the section discusses sensors that are widely available on mobile devices. It then highlights the unique characteristics of those sensors. For the purpose of this thesis, it is important to broaden the understanding on how each technology works, and how it gathers information about the current environment. For this reason, these technologies are discussed below.

Bluetooth was initially developed to provide wireless connectivity within 10 metres without cable connections (Bhagwat, 2001). Bluetooth makes it possible for electronic devices to communicate without a physical connection using the 2.4 – 2.48 GHz unlicensed Radio Frequency (RF) band. This technology uses RF signals to set up a point-to-point and point-to-multipoint connection for voice and data transfer within a 10m radius. Each Bluetooth sensor is assigned a unique address known as a MAC address and is programmed to search for other Bluetooth sensors within its signalling (i.e. coverage) range. The device that initiates the connection is called the ‘master’ and the other device is called the ‘slave’. When there are two or more devices connected, they form a network called a piconet. The number of devices in a piconet is limited to eight due to a three bit device address. There can be multiple piconets connected to form scatternets. This increases the chance of finding more than seven Bluetooth enabled devices in the current environment. Further, Bluetooth offers additional information about the device (e.g. a friendly name offering information on the type of device and its owner), and such information can be useful to make better judgement about the device that the Bluetooth sensor is attached to. In addition, Bluetooth works in indoor and outdoor environments, offering automatic ad-hoc network connection when it finds other sensors within its coverage range, and disconnects when they move outside its range (Bluetooth.com).

Many devices are already Bluetooth enabled (e.g. mobile phones, wireless headsets, car kits, PDAs, keyboards, laptops, navigation systems, printers and mice) and it allows devices to connect to these other devices without a wired connection. As of November 2006, there is an estimated installed base of over 1 billion Bluetooth products in various forms (Bluetooth SIG, 2006). This provides an environment that is densely populated with Bluetooth sensors thus increasing the chances of discovering other Bluetooth sensors. It is also important to note that Bluetooth technology has made significant improvement in recent years and has managed to produce sensors that cover up to 100m radius (msmobiles, 2005). However, Bluetooth’s long range sensors are generally not embedded into mobile devices, but rather used as fixed access points. Thus, they do not necessarily offer high mobility i.e. it can only help mobile devices to sense their environment in relation to fixed Bluetooth access points.

The American Satellite based Global Positioning System (GPS) is the most widely used location technology that dominates the navigation and location systems in outdoor environment (Getting, 1993). It is a line-of-sight technology which requires unobstructed view from the satellite to GPS receiver, and thus, GPS becomes non-functional in indoor systems. A GPS receiver must be able to receive signals from at least three satellites to calculate 2D absolute positioning and at least four satellites for 3D absolute positioning. These positions can be identified on a geographic map if necessary. Further, it provides 'absolute' position which is able to specify its current position in latitude and longitude. GPS technology can be useful when trying to offer long range services and in outdoor places where all the other sensors fail to offer coverage.

Infrared transceivers (IR) have been used in remote controls (e.g. television and garage doors) and mobile phones for many years. It is a line-of-sight technology and fails to work when the signal is obstructed. It is compact and cheap. Its typical range is up to 5m and could be useful for medium range sensing. Sunlight and fluorescent lights interfere with IR signals. IR transceivers have a very narrow beam (within a 30 degree angle) and the pair of communicating devices must be aimed at one another. Further, transmitter and receiver are expected to remain fixed for the duration of the communication and to be within a range of few metres. The 'point-and-shoot' technique used by IR technology makes it difficult for it to sense other IR sensors in the environment. Thus, mobile device users will have to find other IR enabled entities and point in that direction to enable sensing (Priyantha *et al*., 2000).

Barcode technology requires a barcode (a machine-readable printed using dark ink on a light background) and optical scanners. Barcodes are not powered, can be printed on any size, and stuck to almost anything. They are cheap, light-weight and can be sensed or 'read' relatively quickly by scanners. However, as with all technologies, Barcode has its own limitations. It supports relative short range line-of-sight sensing. Further, they can be easily duplicated (Rico *et al*., 2006) and can introduce security issues when used in systems that need to uniquely identify its elements.

Passive and Active Radio Frequency IDentity (RFID) systems consist of two components; a transponder which is known as data carrying device and a reader that is used to retrieve the data stored in the data carrying device. The data carrying device on passive tag does not have its own voltage. It is activated when it enters the interrogation zone of the reader. The power supply to activate the data-carrying device is supplied by the reader. It would be useful to find devices that are only a few centimetres (cms) away. In contrast to passive tags, active tags are battery powered and are capable of covering longer range in comparison to passive tag's often at a distance of just a few centimetres,. Radio Frequency ID (RFID) tags that may be embedded into many everyday objects, providing more localised estimates of position as they pass closely by RFID readers, suitable for finding devices in close proximity. These tags are small, light-weight and relatively cheap. In terms of installation, tags do not have to be setup or configured, hence can be carried around by users and can be stuck to almost anything. In theory, RFID can work in indoor and outdoor environments however in practise this technology is more suited for indoor environments due to its coverage and the tag's physical characteristics (RFIDJournal.com).

WLAN technology is also another type of sensing technology that is commonly found in mobile computing devices allowing access to email and internet. This technology relies on infrastructures (i.e. one of the sensors involved in the connection must be linked to a fixed network), therefore limiting its services to areas defined by WLAN infrastructures (MobileInfo, 2001).

The table 2.1 summarises the characteristics of various sensing technologies discussed in this section. Each technology has its merits and drawbacks and varies in the way it provides support for gathering context.

Technology	Coverage Environment	Coverage Range	Type	Line-of-sight
Bluetooth	indoor and outdoor	up to 10m up to 100m	relative	no
GPS	Outdoor	global coverage	absolute	yes
Infrared	indoor and outdoor	up to 5m	relative	yes
Barcode	indoor and outdoor	few cms	relative	yes
Passive RFID	indoor and Outdoor	few centimetres or few metres	relative	no
Active RFID	indoor and outdoor	up to 100m	relative	no
WLAN	indoor and outdoor	varies (10m, 100m, 200m, 500m)	relative	no

Table 2.1: **Summary of Sensing Technologies**

It can be seen from table 2.1 that some of the technologies work well indoors yet some of them only work outdoors. Further, they offer different accuracy and coverage range, and have different power requirements. Some of these technologies can be easily deployed into environments and entities. For example, RFID tags can be left anywhere without having to worry about hardware and software installation. Whilst each of these sensing technologies independently has the potential to gather context for a variety of applications, they can also be used in concert with one another to offer different types of services i.e. whilst GPS (an absolute location sensor) could be used to detect places and provide information delivery around those places, short range sensors in the same system could be used to provide messaging in relation to entities such as books and desks.

2.4.4 Mobile Development Framework: Tools for Implementation

Two main technologies are available for developing mobile applications that can run on mobile phones and PDAs: Sun's Java 2 Platform Micro Edition (J2ME) and Microsoft's .NET Compact Framework (referred as .NET Compact Framework in the rest of the thesis). Applications based on J2ME are portable across many mobile phones and PDAs, yet this depends on an individual device's capabilities. .NET Compact Framework is a version of .NET Framework that is designed for windows mobile devices such as PDAs and mobile phones. What follows provides a high-level comparison between the two platforms. It discusses their features and limitations with respect to the goals of this thesis.

Both platforms have their own strengths and weaknesses. J2ME outperforms .NET Compact Framework in portability. .NET Compact Framework can only support the windows operating system; however, with Common Language Runtime (CLR) it can be ported to Windows CE and Pocket PC based operating systems. Nevertheless, there are a wide variety of devices that run on non-windows operating systems, such as Symbian (Newby, 2006), Palm (Palm, 2008a) and other vendor-specific operating systems. These are commonly used in mobile phones, and Palm devices. The .NET Compact Framework therefore is limited in its ability to support all devices that use non-windows operating systems. In contrast, Java provides support for all the operating systems mentioned above. Its 'write once, run everywhere' format is useful for mobile application development and it can be ported to devices running on Symbian, Brew and may more. In addition, Insignia's and IBM's runtime environments help to port Java code to wide range of platforms including Windows.

J2ME's cross platform feature has contributed to the development of vendor-specific toolkits. These toolkits vary widely in the type of software and hardware they can support, and have their own merits and drawbacks. Further, there is no standardisation across these toolkits. As a consequence, these toolkits rely heavily on their own device emulators and editing tools. This poses significant problems and challenges when it comes to choosing and mastering these different toolkits. For .NET Compact Framework, Microsoft's Visual Studio can be used as a development tool. Microsoft's Visual Studio provides a run time engine and class libraries for rapid application development. It provides similar models for desktop and mobile application development and makes the transmission process less problematic for the programmers.

Developing and deploying applications can be cumbersome when using J2ME. In comparison to J2ME, .NET Compact Framework is relatively easy to use when it is combined with Mircrosoft's Visual Studio as it offers support for programming languages such as C# and Visual Basic (VB). It also provides windows controls and libraries that help to inherit window's functionality.

In summary, it is difficult to choose an environment based on just technical feature-to-feature comparison. As developers we need to look at the system as a whole; examining this in relation to the target devices, networking technologies, sensing technologies and to some extent the development tools and drivers that are specific to system development. Further, developers need to understand which platform is better suited for their particular development, and is more likely to support the hardware and software chosen for developing the system.

2.5 Existing Context-Aware Approaches

After discussing the technologies for designing context-aware systems, current approaches to developing context-aware systems are reviewed. In particular, the ways in which these systems have been designed to offer support for user mobility and scalable interaction are discussed. This material helps to identify knowledge gaps that are to be addressed by the thesis.

2.5.1 Proximity Based Approach

This section discusses systems that are designed to provide context-aware services based on proximity, a relationship that is created between co-existing entities (e.g. mobile devices). This relationship is initiated through the sensors embedded into their devices and current environment. What follows examines systems that have adopted a proximity based approach, discusses their choice of technologies used, and evaluates their experiences.

Hewlett-Packard's CoolTown project (Kindberg *et al.,* 2000) provides a web-based solution using various sensing technologies such as Radio Frequency Identification (RFID), InfraRed (IR) and barcodes to support the augmentation of objects and places in the physical environment with web resources. It utilises Uniform Resource Locators (URLs) for sensed information and Web pages for entities; its developers have themselves theoretically discussed its use in messaging scenarios (Kindberg, 2002), although the technical aspects of this are not addressed. However, while using CoolTown to support personal messaging is possible, this would prove to be a complex task, as each entity would need to be marked with a URL and configured for users to create messages. In addition, users need to have the facility to update information on the web every time they want to send a message to someone. For instance, web based solutions execute on nodes statically identified by IP addresses and are connection-oriented. Such models can hardly support the deployment of services over highly dynamic ad-hoc networks (Riva *et al*., 2007). Based on this, it could be argued that CoolTown is more suited for applications with a static information space such as a 'curated' environment rather than dynamic information space, as it is limited in its ability to grow and change dynamically to provide services outside a particular environment.

Hummingbird (Redström *et al*., 1999; Holmquist *et al*., 1999) is probably the only system that has used the term 'proximity-sensitive' in the way that this thesis has addressed. Hummingbird devices provided users with an awareness of other Hummingbird users in their proximity that had a predetermined wish to communicate. Hummingbird is a particularly interesting research project in that it has made explicit its interest in 'local interactions' based on proximity as a device for social interaction, albeit on with a strong notion of synchronous interaction to support interpersonal awareness (they call their device an IPAD, or interpersonal awareness device). Notably, the Hummingbird researchers were also interested in integrating people and places into their system, although technologies to support places were relatively poorly developed. The Hummingbird system had a fixed model of proximity with little or no notion of scale, an understanding of proximity as which is determined by the underlying technology (in this case, roughly 100 metres), rather than its more human counterpart of varying levels of space and scale. Hummingbird devices do not depend on an installed infrastructure, making them flexible and open to use in a wide variety of settings. Yet, although proximity was used by Hummingbird as a mechanism to enable interpersonal awareness, and in the research papers that describe it, little is made of the notion of proximity other than its effects on awareness, which is a very limited lens with which to examine this complex notion through.

The proximity based approach has had some notable developments and research findings in the area of gaming, such as Pirates! (Björk *et al*., 2001; Falk *et al*., 2001) and Feeding Yoshi (Bell *et al*., 2006). It appears that proximity has a particular role to play in game-playing on mobile devices as developers attempt to build systems that bring mobile participants into physical contact with one another and with the material environment as a means of enriching the gaming experience (which itself is often relatively impoverished on a mobile gaming platform). Pirates!, for example, has a thing-to-thing, and a thing-to-place model of interaction, although proximity-driven events are fixed as to their meaning, and cue interaction with the game rather than enriching other forms of connectedness. Thus in sensed player-to-place events, players are informed that they are near to an island, whilst player-to-player relationships show that they are close to other players (Falk *et al*., 2001). However, the authors have not discussed its system architecture outside the area of gaming.

Proximity sensing has also been discussed as a service discovery mechanism, in selecting services relating to a geographical area. The ‘open architecture system’ for mobile location-based applications (Jose, *et al.*, 2001) discusses a distance-based model and scope-based model for discovering services. In the distance-based model, the user is able to specify a distance and look for services within that range from its position. A system designed to alert the user when a friend is nearby (Dahlberg *et al.*, 2000) also works on similar principle and looks for friends in a particular area (not specified by distance-based model but rather coverage range of the underlying technology). However this service is only available for people who are willing to disclose personal information by exchanging their profiles.

Similarly, we have already begun to see proximity-sensitive signalling, such as dating profiles held on mobile devices that trigger when apparently compatible prospective mates pass nearby systems (known as 'proxidating' or 'bluedating'). A related service called Serendipity (Eagle and Pentland, 2004) makes use of users' online profiles to instigate serendipitous interactions between co-existing Bluetooth-enabled devices. Yet these technologies are relatively simple in terms of the services offered, and few offer any great degree of user-configurability or dynamism within the interaction or whilst mobile. They are typically used as electronic initiators for face-to-face communication, or communication in another media (e.g. initiating a subsequent web-based interaction), rather than tools for conducting electronic communication through. Another work carried out to support such proximity-based person-to-person interaction is Time to meet face-to-face and device-to-device (Juhlin and Ostergren, 2006). Authors of Time to meet face-to-face and device-to-device focused on face-to-face meeting through ah-hoc sensors such as IrDA, WLAN and Bluetooth. However, they have not explored the possibility of using their architecture outside face-to-face interaction and collaboration.

This lack of support for mobile communicative interactivity utilising proximity is not entirely absent: the Nokia 'Sensor' application (Nokia, 2007) allows a more sophisticated form of proximal interaction than the other short-range sensor-based applications discussed above. The sensor application runs on a mobile telephone, and passively scans the area for other Bluetooth devices running the same application. Users can initiate connections with other Sensor users in the immediate vicinity, and is intended as a social networking tool. The application supports information sharing between users, so that once a connection between devices has been established, users can browse other users' profiles, and messages can be sent between their devices. Notably, the Sensor application is limited to co-present proximal interaction (connections are not persistent over time), and when devices move out of range, connections are lost. The sensor is not intended as a solution to leave messages in places or on people but used a sensed proximal connection to initiate electronic communication. Nevertheless, this is an interesting development, and offers an insight into the value and potential utility that commercial developers are beginning to place on proximity based systems.

Two other systems worth mentioning in relation to proximity are Relate system (Kortuem *et al.,* 2005) and FarCry (Tennent *et al.,* 2005) frameworks. Relate system extends mobile computing devices with the ability to establish their spatial relationships through a purpose built USB dongle and specialised widgets. Although, Relate system manages to provide accurate spatial information on co-located devices its use is limited due to its purpose built hardware i.e. the design proposes yet another sensor for delivering a new service to users. In contrast to Relate system, FarCry uses exiting WLAN sensors to proximate and spread information. FarCry relies on face-to-face connection to proximate devices and spread files to other mobile devices. As a result a sender's device in FarCry system has no control over file delivery beyond the immediate vicinity. A device in FarCry system copies files directly to other connected devices, thus, FarCry presents a serious security risk to mobile device users. In addition, FarCry support is limited to the areas covered by WLAN networks.

The approaches presented above have not been intended to understand proximity or its novel characteristics. They were mainly developed for supporting context-aware services in a specific application domain (e.g. tourism, gaming, messaging). Further, the researchers of these systems have done very little to explore their architecture to provide coverage outside their domain of interest. As a result, there are significant knowledge gaps in understanding the concept of proximity and the novel characteristics it could offer to context-aware systems.

2.5.2 Relative Location Based Approach

Relative location identifies mobile devices in relation to static entities embedded in places. Many systems have adopted this approach to discover mobile devices and deliver services in relation to those locations. For example, Floating Note (Multaharju *et al*., 2004), PlaceMail (Ludford *et al*., 2007) and E-graffiti let users to see their current locations, leave messages in that location and view messages left by other users who have visited that location. However, they are all WLAN-based systems and thus only work when WLAN is present. Further, their location is determined by the WLAN access point the user is currently connected to, and the granularity of the location information is the size of the cell, something that is invisible to users. Although the authors of the Floating Note system have discussed possibilities of achieving accurate location identification following the method of using signal strengths from several access points (Seppanen, 2002), they do not discuss the possibility of supporting scalable interaction where messages can be delivered in different levels of proximity.

In the scope-based model discussed in Jose *et al*'s (2001) open architecture for mobile location-based applications, a user is able to discover services when he or she is located within the service scope. For example, if a particular service is only available in meeting room RM303, then the user will have to be in that room to access that particular service. Further, the scope-based model describes how to delimit areas for service delivery, each service is assumed to have an associated scope that specifies the physical range in which it should be available. This is a useful feature for targeting an area for information delivery rather than delivering it everywhere. In addition, they have highlighted their interest in delivering services in different levels of spatial specificity such as a building, a room or a desk. However, they have not looked at this outside the areas covered by their system infrastructures.

In addition to the above single sensor systems, there are also multiple-sensor based systems. Urban tapestry (Lane, 2003) and the Mobile Bristol project (mobilebristol.co.uk) are examples for supporting context-awareness through multiple sensors. Both are designed to support ‘public authoring’ in different environments. They both allow users to map and share local knowledge and experience with other users and provide services through sensors that are embedded in the environment (e.g. static sensors). These systems provide a mobile location-based platform to connect users to places, allowing users to author their stories and embed them in places. They do not support communication directly between two people: it is possible for location information to be derived from a remote centralised system that could then deduce that user devices are in the same location, but this location is not derived from a peer-to-peer connection. These systems too rely on pre-existing infrastructures. Several other systems such as Microsoft’s Easy Living (Brummit, *et al.*, 2000) and Interactive Workspaces (Johanson *et al.*, 2002) have also adopted such relative location-based architectures to provide services.

The mobile phone network is one of the most common infrastructures that are currently being used for locating mobile devices and offer services such as finding the nearest restaurant, hotels traffic alert, etc. In theory, mobile phone systems are capable of using cell triangulation techniques to obtain the location of a mobile device. However in practice, the operators are still using the Northing and Easting of the serving cell as the location of the mobile device. The mobile device could therefore be anywhere in the coverage area of the serving cell. In a typical rural area, a cell is designed to cover around 10-20 km and in urban area up to 200-500m. The location information provided by mobile phone network is not accurate and at a highly variable scale, and is therefore not precise enough for most location-based systems to provide the level of services expected by the consumers. This technology however works in both indoor and outdoor environments without any major problems, although the accuracy of this system is far from adequate to support indoor location.

Mobile Ward (Skove *et al*., 2006) is a prototype that helps to explore context-aware features in a hospital environment. In this, people, places and time are modelled into the system to enable context-related information delivery to medical professionals. The Mobile Ward prototype sends information according to the physical location of the user. For example, when a nurse walks into a ward, the system automatically detects their change of physical location and provides patients' information related to that particular ward, and when the same nurse gets closer to a patient's bed in that ward the nurse is presented with information relevant to that patient. Although this prototype has modelled people into the system it has confined those people within places (locations) and static entities (beds). According to its published material, Mobile Ward delivers information to its users based on where they are, and there is no indication on whether it supports information delivery based on who is nearby. Such a form of information delivery would be required when a patient needs to be examined or treated outside his or her ward or bed. For example, a nurse may need the medical notes when he or she is with the patient in the treatment room, and this room could well be outside the ward. This illustrates how identifying people and objects in relation to a location is not sufficient to cover all aspects of movement.

All the above systems rely on infrastructure and therefore still have certain element of fixedness in them in that mobile devices can only be discovered in relation to static entities. As a consequence, their architectural features fail to support communication in highly mobile and dynamic environments where all interacting entities could be mobile.

2.5.3 Absolute Location Based Approach

Location based approach provides devices with specific coordinates that can be mapped on to a geographical map. GeoNotes (Espinoza *et al*. 2001) is a system that was designed to allow the information space to grow, expand and develop with users rather than maintaining a static information source that was created by developers of the system. Users are allowed to provide, update and remove information. This facilitates information flow in both ways, i.e. user-to-system and system-to-user. GeoNotes uses WLAN technology to connect a user's PDA or laptop to a central server to store and retrieve location-based information. It does not rely on any particular type of location technology but uses the location technology that is available on the device (e.g. GSM, GPS). This allows GeoNotes users to automatically detect their current geographic position in the network and write 'tags' and electronic graffiti at that particular place (Persson *et al*., 2002). Places are chosen and defined by users, as a response to their mobility needs. This system limits its services to locations and relies on pre-existing infrastructures.

Another related system that uses absolute location is ComMotion. This system was developed to provide some flexibility to users in selecting places for communication, and allow to write messages to others and personal reminders. It uses GPS co-ordinates technology for locating mobile users. It learns users frequently visited locations and prompts them for a place name. The named locations in comMotion can be tagged with messages and later delivered to users when they are in that vicinity. ComMotion also supports subscribed information such as news headlines, weather and other local information. This system enables users to provide information to the system and supports marking to certain level. This is an improvement from static location systems; however comMotion cannot solve the issue of mobility in its entirety. It focuses on marking locations rather than any entity (even mobile entities) and limits its choice to frequently visited places. Point-to-GeoBlog (Robinson *et al.,* 2008) marks points to support user generated content creation. This allows users to mark points by simply pointing at their area of interest with their PDAs, refine the targeting by tilting and clicking when the marker is positioned over their desired target location. Like comMotion, this also uses GPS to determine a user's location, and thus is suited to outdoor environment due to the inherent limitations in GPS technology.

Another GPS based system is PlaceMemo (Esbjörnsson and Brunnberg, 2001), which lets users to take an active part in providing information to the system. The user adds information to the system in the form of voice messages. In this, users are allowed to attach voice messages to locations. These attached messages are then delivered to recipients when they pass by that location. This is considered as user-to-user messaging, maintaining a personal virtual information space for each user, designed especially for the user's private context. Information elements in the spaces are copied from one user space (sender's), into another user's (recipient's) space. Sharing memos in PlaceMemo are serviced by copying memos directly to user's information space on the assumption that the memo is relevant and important to that recipient.

Other systems such as ActiveBadge (Want *et al*., 1992; Harter and Hopper, 1994), Active Bat (Harter and Hopper, 1997; Harter *et al*., 1999) and Cricket (Priyantha *et al*., 2000) were developed to facilitate indoor location with higher level of accuracy compared to GPS and mobile phone network triangulation. The Active badge was one of the early indoor location systems that used infrared (IR) technology to transmit data. In this system, people wore small computing devices known as badges and each one of these had an IR emitter which sends unique pulse signal in a defined time interval. Purpose-built sensors were placed in every room to detect these signals. IR is a line of sight technology and therefore the signals are confined within each room. The location of the user is identified by the sensors located in each room i.e. the location of the sensor that received the signal is the location of the user. This form of context-awareness is useful for tracking, delivering information in relation to various locations and not beyond locations. Active Bat (Harter *et al.*, 1999) is another location-based system which uses the principle of triangulation (position finding by measurement of distances). Each device needs three or more such distances, to determine the 3D position of each Bat. In this, the Bat is a transmitter that is attached to the object that needs to be located. By finding the relative positions of two or more Bats attached to an object, it is possible to calculate its position.

The Cricket indoor location system uses a combination of Radio Frequency (RF) and ultrasound technologies to provide location information to attached host devices. Wall and ceiling mounted beacons placed in buildings broadcasts information via RF channel. Using RF broadcast, the beacon transmits a concurrent ultrasonic pulse. Listeners attached to beacon sensitive devices listen for RF signals, and upon receipt of the first few bits, listen for the corresponding ultrasonic pulse. When this pulse arrives, the listener obtains a distance estimate for the corresponding beacon by taking advantage of the difference in propagation speeds between RF (speed of light) and ultrasound (speed of sound). The listener uses algorithms that correlate RF and ultrasound samples to select the best correlation. Even in the presence of several competing beacon transmissions, Cricket achieves good precision and accuracy quite quickly. Although these indoor location systems are capable of offering the precision required by indoor messaging systems, they need purpose built infrastructures to offer services which limits services to locations within areas covered by system's infrastructures.

One of the recent projects worth mentioning is Place Lab (LaMarca *et al*., 2005). This is a research project that attempts to solve the coverage issues surrounding wireless-based location estimation. Place Lab predicts location using the known positions of the access points detected by the device. The positions of these access points are retrieved from a database cached on the same device. Place Lab uses GSM Base Stations and fixed Bluetooth devices as well as 802.11 access points. The Bluetooth devices improve Place Lab's accuracy when they are available. In residential and urban settings with GSM coverage and moderate 802.11 set up, Place Lab produces location estimates with 20-25 metres of accuracy. Place Lab addresses both the lack of ubiquity and the high-cost of entry of current approaches to location. Yet Place Lab is different from most of the other coexisting systems as it allows commodity hardware like notebooks, PDAs and cell phones to locate themselves by listening for radio beacons such as 802.11 access points, GSM cells, and fixed Bluetooth devices that already exist in the environment. All these beacons have unique or semi-unique IDs, for example, a MAC address. Clients compute their own location by listening to one or more IDs, looking up the related beacons' positions in a locally cached database, and estimating their own position in relation to the beacons' positions recorded in the database. Based on this, developers are allowed build their own location-based applications. Place Lab attempts to provide location, based on predefined database and various technologies that can sense the device presence in the environment.

Although Place Lab utilises existing infrastructure and sensor enabled devices in the environment it requires a database to calculate Clients' positions. The information (beacon positions) in the database has to be collected and stored before they can be utilised by the Place Lab Clients. Place Lab authors have highlighted the accuracy and availability problems of Place Lab database in Borriello *et al* (2005) and have discussed how every day mobile devices can be utilised to minimise these problems i.e. every day mobile devices with GPS can identify beacons and record their positions in their environment to provide up to date information to Place Lab database. However, Place Lab architecture always needs a predefined database to calculate Clients positions. In addition, Place Lab fails to provide support for marking any informationally-interesting mobile entities.

Second Generation (GSM) and Third Generation (UMTS) mobile phone network based LBS provide personalised information to subscribers based on their current position. A mobile device's location can be identified using either the cellid technique or using additional information available in the network such as timing advance (TA) and network measurement reports (NMR) (3GPP, 2004). This information is available for all handsets. Currently advanced techniques such as Enhanced Observed Time Difference (E-OTD) and Assisted GPS (A-GPS) are being introduced in new handsets. The accuracy and speed of location estimation of A-GPS is improved by the information provided by GSM network. Based on this location mobile phone can download the anticipated position of the satellites allowing the handset to lock on to GPS in seconds. However E-OTD and A-GPS require more complex and expensive handsets to implement such systems and therefore it has not yet been adopted by developers to provide services to general public.

As can be seen from the above discussion, most existing location-based systems rely heavily on infrastructures and process intensive (the exact location calculation or query database for location data) solutions. Yet as discussed before, for many forms of context-aware services, location is not particularly useful and moreover, may not be always necessary. Further, most location-based systems do not allow context-awareness to take place in relation to other mobile entities.

2.5.4 Useful Architectures for Context-Awareness

This section looks at some of the system architectures that are relevant to the identified research problem. These systems have used unique characteristics of different technologies to incorporate different set of features into their systems. These systems are explored in order to learn from their choice of technologies.

The majority of commercial location-based systems rely heavily on GPS and GSM. However, the indoor systems cannot rely on these technologies as they are likely to require fine grained sensor discovery. Consequently, indoor systems are usually designed using medium and short range sensors such as Bluetooth, RFID, IR and ultrasonic. Transmission range (i.e. coverage range) for these individual sensors varies and they are put to very different use by researchers, offering discovery in different granularities of scale. A Bluetooth based indoor positioning system (Forno *et al*., 2005) is an example of Bluetooth being used for indoor positioning: it uses two different cyclic powers to estimate the distance from the sensor (i.e. under 5m or 8-10m). The author describes that this can be a complex architecture when lot of sensors are involved and states that the data collected by these ad-hoc sensors are sent to a remote centralised positioning system. Although Forno *et al*'s (2005) Bluetooth based indoor positioning system offers precision up to 1.88 metres, it does require a purpose-built Bluetooth infrastructure. This becomes an issue when people want to communicate across wider areas, leading to questions, such as how much area can be covered by these sensors and what happens when the communication takes place outside this area? However, this architecture helps to build a richer picture of how Bluetooth can be utilised to provide support for services outside the area covered by infrastructures and to learn from its limitations.

Moving on from this, other systems have incorporated multiple technologies to provide services in outdoor and indoor settings. Place Lab is an example for this. It uses location-inferred discovery to find devices in the environment. However, these systems cannot be used in the first messaging scenario (section 1.2.2), as there is no guarantee that the subsequent meeting will take place in a known location or place. The SLAM (Scalable Location-Aware Monitoring) project intends to support wide range of tracking and controlling applications (Priyantha *et al.*, 2000 and 2001). Although this is not tailored for messaging services, the system uses technological convergence to resolve the restricted coverage limitations in location monitoring systems by using multi-sensor discovery from different technologies, such as GPS, RF and ultrasonic sensors along with RFID, to provide coverage across various environments. The clear distinction between SLAM and what is discussed in this thesis, is that SLAM uses purpose built ultrasonic beacons to offer high accuracy in indoor environments, whereas this thesis attempts to base its design on existing wireless sensors that are already embedded into mobile devices and users' environments. Thus, the similarities between SLAM and the system discussed in this thesis are more to do with the nature of sensor discovery than the purposes to which the systems are put.

2.5.5 Current Systems' Limitations

The majority of the system architectures discussed in sections 2.5.1 to 2.5.4 are predominantly service-oriented and have not been designed to be 'open' in the way they adapt to different sensing technologies, environments and entities (i.e. people, places and objects). They are typically designed to support a particular kind of service within an environment (indoor or outdoor) using a particular type of technology. The technical characteristics of the underlying technology influence the functionality of the system as to where it will work (indoor, outdoor) and how much precision it can offer, and as a result, single technology-based systems are limited to a particular type of environment and proximity range. Further, service-oriented approaches are generally tailored to specific domains and hence are not general purpose, and cannot be used to offer different kind of services (Rahmani *el al.,* 2006). Therefore a more general approach is needed to design complex systems such as context-aware systems. The research in the past has done very little to utilise the proximity information available from sensors. What follows is a set of reasons why these systems cannot be used to support context-aware services in mobile and dynamic environments without significant modification to their architectures.

Context-aware services demand for a system that is 'always on' and available for providing services where and when they are needed. When recipients enter an area, they have no knowledge on whether information is waiting for them (context related message). Thus, context-aware systems rules out the option of 'turn on' when required or log into the system to access information. In order to keep the systems turned on all the time, mobile devices need to have longer battery life. Additionally, people often move around and take their devices with them and may want to leave information for others in relation to static or mobile entities. This means that technologies should be able to support highly mobile environments providing coverage wherever the service is needed. Currently, no system is able to support services everywhere; systems such as E-graffiti and Floating Note are only able to work indoors where WLAN technology is present, and comMotion is only able to offer its services in outdoor environments where GPS can work. This shows that single technology solutions or infrastructure dependent systems are not going to be effective to support discovery and services in and across various environments.

Mobile Bristol, Urban Tapestry, CoolTown and SLAM are some of the systems that have used multiple technologies. However, Mobile Bristol and Urban Tapestry require purpose-built infrastructures whereas CoolTown relies heavily on a web model and wireless sensors within an area. The SLAM project has tried to tackle the coverage issue by combining standard GPS receivers with custom RF and ultrasonic beacons. SLAM also proposes the idea of tagging objects to overcome the practical problem of attaching purpose-built listeners to all of the objects in users' environments. Despites its efforts to resolve the coverage issue, SLAM still needs a purpose-built infrastructure to locate users' devices. This introduces infrastructural overheads and maintenance issues. Further, it identifies devices in relation to their location, and thus limits mobility. Nevertheless, comMotion's marking of places and SLAM's multiple technology approaches relating to coverage issues are considered useful to address coverage issues in this thesis; incorporating different entities and technologies increases the chances of finding at least a technology (sensor-enabled entity) in the environment.

The issue of scalability has been raised by various messaging systems (e.g. Floating Note and PlaceMemo), providing support as people often wish to leave messages at various levels of scale, ranging from making a note on a small object (e.g. a book, a desktop PC), to entities or places at larger scale (e.g. a room, a person), or to post messages over wider areas (e.g. campus, airport). Yet, at the time this research was started, most single technology systems were designed to offer services for a single range. According to the author of Floating Note, its granularity of the location information is delivered within the cell's coverage area. A user wanting to limit message delivery to few centimetres is forced to use up to 10 metre radius delivering messages before they are needed. In the same way, if the user wants to deliver a message using Floating Note to users in 50 metre radius then there is a possibility that the message might not be delivered or out of context. This is an area of research that has not been addressed by current systems.

In addition to the above issues, there are often maintenance problems for embedded sensors. The majority of existing systems require sensors to be installed and set up by professionals before they can be used. The ideal situation for this is to allow sensors to be added dynamically and removed when they are no longer required. Systems that provide entity discovery based on an infrastructure and purpose-built sensors are difficult to maintain as these sensors have to be individually installed and incorporated into the system.

In summary, researchers so far have mainly focused on location-based and infrastructure-dependent or application-specific systems. Even the ones that have adopted a proximity-based approach have failed to address the potential benefits of proximity sensing. As a consequence, these systems are limited in their ability to adapt to highly mobile and dynamic environments, in particular, those which relate to device and entity mobility. Further, they have generally focussed on the single level of interaction with no notion of scale, focusing only on the presence of discovered entities. Thus, limiting spatial association (i.e. only a single relational distance between sensors could be associated with a service) and its communicative affordances (i.e. a single level of proximity affords less variations on content interpretation).

2.6 Summary and Conclusion

The chapter has discussed various technologies and published materials relating to existing proximity-sensing and location-based (absolute and relative) systems. This discussion has highlighted some gaps in the research and pointed out why current context-aware systems are struggling to provide support for ubiquitous coverage, mobility and scalability. In addition, it helped to identify specific problems which currently need to be addressed to find support for mobility and scalability. Based on these findings, the rest of the thesis discusses the development stages of a novel system that provides potential solutions to problems that have prevented existing systems being used to deliver scalable context-aware services in highly mobile and dynamic environments.

Chapter 3: A Proximity Based Architecture for Context-Awareness

3.1 Introduction

This chapter presents a set of requirements that helps to define the characteristics for a context-aware system to enable scalable interaction in mobile and dynamic environments. The chapter elucidates how these individual context-aware system requirements evolve into design considerations and later into decisions that lead to a proximity-based approach. Following on from this, the chapter describes the proximity-based approach, introduces a Proximity-Sensitive System Architecture, and explains how the individual elements of this architecture interlink and interact with each other to gather the information necessary to support context-awareness.

3.2 Context-Aware System Requirements

A reflective critique of the relevant systems was provided in Chapter 2 to examine why current context-aware systems in general are not able to adapt to mobile and dynamic environments. The knowledge gained from this helped to understand the main design constraints present within existing Context-Aware Systems and identify the characteristics that such a system should encompass in order to support scalable context-aware interactions in mobile and dynamic environments. This section describes these coverage, mobility and scalability characteristics and explains why they are important to any Context-Aware Systems. Finally, it lists and discusses a set of requirements that help to define these desired characteristics.

3.2.1 Maximising Coverage

The thesis argues that there is very little use in designing a context-aware system if it cannot provide coverage to offer services *where* people want them to work. As noted in the previous chapter, this coverage issue has been acknowledged by a number of researchers and attempts have been made to extend system coverage to wider areas (e.g. LaMarca *et al.,* 2005; Chin *et al.,* 2005; Howard *et al.*, 2002). The intention of the thesis is to extend system coverage beyond infrastructurally-defined areas through the use of a variety of entities (including not just static but also mobile entities) and technologies (i.e. multiple sensor technologies). Below, are three requirements R1, R2 and R3 that will enable context-aware systems to maximise coverage.

R1: Allow sensing to take place in a wide range of environments

This requirement focuses on providing system coverage in a wide range of environments without limiting to a particular type of environment such as indoor or outdoor. This will allow users to access and leave their context related information and messages in both indoor and outdoor.

Mobile device users and entities with which they interact move across a wide range of environments (including both indoor and outdoor) and communicate with other users and information sources. This form of user and entity mobility poses additional challenges to context-aware system designers as it demands for coverage in and across various indoor (e.g. museums, supermarkets, airports) and outdoor (e.g. car parks, motorways) settings. The majority of current context-aware system approaches are unable to provide coverage for different environments.

R2: Allow sensing in relation to different type of entities

The focus of this requirement is to allow entities to be sensor enabled so that they too can take part in providing coverage. In mobile and dynamic environments, there is more chance for users to move outside areas covered by the system's infrastructure. However, whilst on the move they may come across a variety of entities; current approaches often do not enable users to incorporate these entities. To address this limitation, the design supports the augmentation of these entities to take context-aware services beyond areas covered by system's infrastructure.

R3: Adapt to different technologies

The objective is to increase coverage by allowing the system to discover sensors that belong to different technologies. By doing this, we aim to increase the chance of finding at least one of the technologies in a user's current environment to provide the coverage.

Technology-specific context-aware systems do not have widespread applicability as their functions are limited by the characteristics of underlying technology (Mitchell, 2002). In addition, the services they offer are only available to systems supporting the particular technologies employed.

The above three requirements (R1, R2 and R3) will allow the design to take advantage of a wide variety of technologies integrated into mobile entities, and embedded into the physical environment. In this, the sensors integrated into mobile devices are not restricted to a particular environment, will have the potential to provide coverage even outside the areas covered by embedded sensors, offering coverage beyond a system's preconfigured infrastructure.

3.2.2 Supporting Mobility

In terms of supporting mobility, existing context-aware approaches have primarily focussed on two types of mobility. The first focuses only on the mobility of the user, who may want to move around and interact with static entities such as desktop computers in the environment. The second concerns the mobility of the user and the mobile device, for example, a mobile device user moves around and uses his or her device to interact with static entities in the environment. This form of mobility is supported by many context-aware systems. However, this mobility alone is not enough to address the first messaging scenario discussed in Chapter 1, i.e. to support context-aware interaction in relation to mobile entities. In this, not only users and their devices are mobile but the entities with which they interact may also be mobile. Based on the device mobility, and device and entity mobility types discussed in Chapter 2, the thesis derives two further requirements, R4 and R5, for the system design.

R4: Allow mobile devices to discover and mark static entities

This requirement not only concentrates on enabling mobile devices with the ability to discover static entities (e.g. static objects and places) but it also focuses on marking those entities. It allows mobile devices users to find entities around them through their mobile devices, and mark entities of their choice for tagging information on them.

R5: Allow user's mobile device to discover and mark mobile entities

This requirement is different to R4 as it focuses on discovering and marking mobile entities. Marking provides users with the opportunity to make a choice on where they want to leave their information. Requirement R4 will allow users to interact in predefined places determined by the system's infrastructure and, where necessary in relation to any static entity (e.g. a marked door). This is an improvement over the current context-aware approaches, however not enough to support the kind of mobility and dynamism introduced by mobile and wireless technologies (see Section 1.2.2).

At face value, R5 seems a fairly simple requirement to fulfil, as a large number of mobile entities (e.g. mobile phones, PDAs, laptops, earphones, GPS receivers and cameras) are already sensor-enabled and can be easily discovered by users' mobile devices, and therefore can be marked for supporting interaction. Yet, not all informationally-interesting mobile entities are sensor-enabled, and for example, books, files and DVDs are unlikely to have a sensor or power source. R5 is included to overcome this and provide support for the discovery and marking of mobile entities that have no means of providing their own power requirement.

3.2.3 Supporting Scalable Interaction - Scalability

PlaceMemo (Gustafsson, 2005) has highlighted the importance of determining perimeters for triggering messages. It tries to provide information in advance of the user encountering the marked entity so the user can decide what measures to take before it is too late and he or she has passed it. This highlights that scaling and delivering information within that scaled area is important for context-aware services. For example, in a context-aware personal messaging system, a message reminding someone to borrow a book from the local library will require larger messaging area compared to a reminder left on a desk to check for some information on the internet. To achieve this characteristic and offer interactions at three different levels of scale, the following three requirements R6, R7 and R8 are included. These three requirements can be put to very different uses – the reasons why a user would tag a space to a range of a few centimetres are likely to be different to why they would do so where the range and accuracy of the tag is in the order of a radius of 10, or 100 metres. Whilst each of these levels may be used independently, they can also be used in conjunction with one another to provide scalable context-aware services.

R6: Support centimetre-level granularity

This requirement is included to enable short-range sensing on mobile devices, to discover entities within a few centimetres of a mobile device.

R7: Support up to 10 metres granularity

This particular range is included to provide medium-range sensing on mobile devices, to discover entities within 10 metre of a mobile device.

R8: Support greater than 10 metres granularity (e.g. 100m, 500m and 1km)

This requirement is incorporated to identify entities when the mobile device is tens of metres away from the marked entity. This is generally suited for outdoor environments where fine-grained information delivery may not be required.

Utilising a single level of proximity is communicatively limiting for reasons of access (it provides fewer opportunities for discovering a proximal relationship), spatial association (only a single relational distances between sensors could be associated with a service) and its communicative affordances (a single level of proximity affords less variations on content interpretation). Having three levels of distance will allow users to choose areas more appropriately for delivering their information.

The above requirements are used as input for designing the system. They will help to decide on the approach for enabling maximum coverage, mobility and scalability.

3.3 Meeting the System Requirements

Based on the requirements specified above, the remainder of this section explains why more appropriate context types (see Section 1.2.1) and technologies are required to design useful and flexible context-aware systems, and provide a rationale for the choices made, before proposing a proximity-based architecture. The review of relevant technologies and critique of the existing context-aware systems discussed in Chapter 2 are used as an aid to make design decisions: type of context, sensor information and technologies to be used to design a Proximity-Sensitive System Architecture.

3.3.1 Type of Context to be Used

Context-aware applications are limited by the context type they use to provide services. It was highlighted in Chapter 1 that the majority of the context-aware systems are based on context type location and therefore they rely heavily on entities that are static (i.e. fixed to a location). As a result, these entities are not free to move around to provide coverage elsewhere. To overcome this limitation, and offer support for mobility and scalability a different context type called 'proximity' is examined and explored in this thesis.

Proximity is based on spatial relationship between entities. Therefore it does not require network entities to be static, it allows entities to move around and still discover each other irrespective of their location. This characteristic of proximity helps to support mobility by extending coverage beyond a particular environment or area. For example, consider *ad hoc* network elements that need not be physically connected (i.e. not hardwired together). They have the flexibility to move beyond the reach of wired connections and still provide coverage wherever they may be. Further, it makes designs easier to evolve as new sensors and devices appear in the system's surrounding environment i.e. they become part of the system by simply being in the environment rather than being physically integrated into architecture through wires or by installation.

In addition, proximity sensing discovers spatial relationships between entities, how *close* is a mobile device (or a user) to an entity. This characteristic (i.e. *closeness* between entities) could be exploited to offer different levels of spatial granularity. For these reasons, context type proximity seems appropriate for supporting maximum coverage, mobility and scalability compared to location.

3.3.2 Gathering Context Information Through Sensing

Once the decision was made to use proximity, it was necessary to decide whether to utilise low level or higher level context information to provide context-aware services. Dey *et al.* characterise low-level information as data that is readily available from sensors and high-level information as interpreted data that is driven from low-level data. For example, if a context-aware application is taking details such as unique identification number of RFID tag as an input, then it is utilising low-level information, but if the application is taking details such as whether a meeting is taking place in a room (i.e. interpreted from co-presence of sensors), then that application is utilising higher level information. In order to support the kind of services discussed in the scenarios in Chapter 1, the application must be able to determine a user's presence in relation to an entity in his or her surrounding environment. This can be determined using low-level information provided by various sensors. However, there are a wide range of sensors which gather different types of information about the sensors in their environment (e.g. unique sensor identity number, position, etc.) and difficulty arises when wanting to incorporate more than one type of sensor into the design, each of which will provide different types of data and a range of values.

Pascoe (1998) identified difficulties in developing software that can capture context using a variety of hardware, translate into relevant formats, interpret and utilise it to provide meaningful information. Several systems have thus been developed since then to simplify the design process, providing solutions in the form of toolkits (e.g. Dey, 2000) or architectures (e.g. Hong and Landay 2001). Further, researchers like Gellersen *et al.* (2002) have described that it is beneficial to use multiple, comparatively simple and diverse sensors to access contextual information, as this will offer an opportunity to gather different information about the environment, and thus will help to provide different types of services. In addition, recent literature findings suggests that wireless and sensing technologies have become pervasive enough or are likely to become pervasive in users environments through sensors embedded in mobile devices (see Bluetooth SIG, 2006; Kindberg *et al.,* 2002) and within environments (LaMarca *et al*., 2005; Schmandt *et al*., 2000) to provide coverage. Indeed, these wireless and sensing technologies provide a ready-made platform for designing systems that will help to gather a wide range of low-level information about context, and thus support a wide range of services.

3.3.3 Technologies for Gathering Contextual Information

This section helps to identify suitable technologies for extracting contextual information through proximity sensing. Context information can be gathered using a single technology (e.g. Espinoza *et al.,* 2001) or multiple technologies (e.g. LaMarca *et al.,* 2005). Chapter 2 examined the two strategies by critiquing systems that have adopted these different strategies. Further, it discussed various technologies for sensing and arrived at the conclusion that a single technology solution alone would not be sufficient to meet the requirements listed Section 3.2 (R1 to R8). Adopting a multiple technology solution is therefore an appropriate strategy for this design, i.e. a solution that exploits technological convergence to bring together mobile devices and multiple sensing technologies. The rationale for the choice of technologies is discussed below.

First, the focus is on technologies that can help to maximise coverage. Three requirements (R1 to R3) were introduced by this characteristic: to allow sensing to take place in a wide range of environments, to allow sensing in relation to different type of entities, and to allow the system to adapt to different technologies.

Current GPS technology is mainly limited to outdoor location although it is quite useful to utilise this in applications that use geographical coordinates (e.g. Abowd *et al.*, 1997; Marmasse and Schmandt 2000) or the area around such coordinates. Technologies such as Bluetooth, RFID, Infrared (IR) and Barcodes can technically work almost anywhere independently. However, RFID and Barcodes are more suitable for indoor environments where fine grained positioning is needed (e.g. Yun-Maw *et al*., 2005), rather than outdoor environments that seem likely to require coarse grained positioning. In addition, their physical characteristics, in that they can easily be lost or damaged is not generally practical for outdoor environments. IR is a short-range line-of-sight technology, and both transmitter and receiver must be almost directly aligned for it to communicate. For this reason, it cannot be used to provide coverage in wider areas without deploying multiple IR sensors. This leaves Bluetooth as a better candidate for providing indoor and outdoor coverage within a limited range (10m).

Second, the focus is on technologies that can support mobility requirements (R4 and R5). Bluetooth sensors can be theoretically embedded into mobile and fixed entities: allowing sensing of mobile (e.g. PDAs, laptops) and static entities (e.g. printer, desktop PC). However, Bluetooth sensors need to be installed, configured and connected to a power source. Consequently, Bluetooth cannot be used for sensing entities that have no means of providing their own power (e.g. books, office doors and files). Contrastingly, passive RFID tags and Barcodes undoubtedly offer better support for such entities, as they support relatively cheap 'fit-and-forget' placement offering support to sense entities without power.

Finally, the technologies are reviewed with respect to scalability requirements (R6 to R8). In this respect, a relational association can be naturally derived from spatial distance when entities sense each other. Each sensor has a unique set of characteristics and supports sensing in different spatial distance and therefore taking advantage of technological convergence can help to support scalability: short, medium and long range proximity sensing.

In order to sense the presence of context messaging in relation to entities such as books, keys and DVDs, passive RFID and barcodes technologies would be more appropriate (e.g. scenario 2 in section 1.2.2) as they support few centimetres coverage. One of the main advantages that RFID has over barcodes is that it is not a line-of-sight technology. Therefore, users do not have to make sure that the reader and RFID tags are directly aligned for discovery. In addition, RFID is more secure, as each tag is identified by a Unique Identification Number (UIN); the advantage of this tag is that the data on the chip that uses sophisticated algorithm techniques that cannot be duplicated or manipulated easily like Barcodes (RFIDJournal.com). This offers an advantage when supporting security related proximity-sensitive services.

Short-range passive RFID however offers very little support for the first messaging scenario (see Section 1.2.2) as it requires a wider coverage, and it would be more appropriate to use up to 10 metre medium-range for this kind of interaction. Commonly used Bluetooth and some WLAN technologies are suitable for this range of operation. One useful advantage of Bluetooth technology is that it can always operate independently, without any fixed network connections. Thus, Bluetooth offers better support to mobility than WLAN with respect to the kind of services discussed in Chapter 1 i.e. Bluetooth can discover static and mobile entities at the same time without being connected to a fixed network point.

Another possible interaction setting involves outdoor sensing, in places like car parks or motorway junctions, demanding even longer range than a 10 metre radius (e.g. 100 metres). Although mobile phone networks and long-range WLAN already offers services in this range, they have their own limitations. The mobile phone network offers very little support for developing applications on top of them without getting help from mobile phone operators. When the research was first started, WLAN offered wireless support in public places and inside buildings with limited support for outdoor remote areas (Schmidt and Townsend, 2003). In general WLAN is set up to work in infrastructure mode, allowing discovery through fixed access points. In addition, WLAN only works in ad-hoc mode or infrastructure mode at any given time and therefore cannot fully support mobility. It is worth noting that Bluetooth now offers long-range support through fixed Bluetooth points and USB dongles for Windows xp devices. However, long-range Bluetooth was developed very recently for consideration in this thesis. This left the design space with GPS as the only technology available with wide enough coverage for supporting long-range proximity sensing. Notably, GPS provides information about the location and not proximity. GPS has therefore been selected as its location information helps to provide proximal sensing i.e. identify devices when they are within a specific range from its current GPS coordinate.

Technology	**Maximising Coverage**			**Supporting Mobility**		**Supporting Scalability**		
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**
GPS	◖	◖	⊗	●	⊗	⊗	⊗	●
Bluetooth (Range < 10m)	●	●	⊗	●	●	⊗	●	⊗
WLAN (Range <10m)	●	◖	⊗	●	◖	⊗	●	⊗
RFID (Range up to few cm)	◖	●	⊗	●	●	●	⊗	⊗
IR	◖	●	⊗	●	●	●	⊗	⊗
Barcodes	◖	●	⊗	●	●	●	⊗	⊗
Bluetooth (Range < 100m)	●	●	⊗	●	●	⊗	⊗	◖
WLAN (Range < 100m)	●	◖	⊗	●	◖	⊗	⊗	◖

Key: ● = Full Support, ◖ = Limited Support, ⊗ = No Support

Table 3.1: System Requirements Vs Technologies

Table 3.1 summarises sensing technologies with respect to system requirements (R1 to R8), their suitability for maximising coverage, supporting mobility and scalability. This comparison allows to select technologies for creating a suitable design for proximity sensing. In summary, the final design decision was made to use multiple sensing technologies to gather sensor data that provides information on proximal relationship between entities.

3.4 Proximity-Sensitive System Design

The key objective of the design was to create a system architecture for context-awareness that provides the characteristics of proximity through the use of existing and commercially available technologies, mobile devices, infrastructures and *ad ho*c sensor networks. More specifically, the design focuses on creating a system that can work on top of existing infrastructures and *ad hoc* sensor networks without the requirement of purpose built networks. This approach provides a more practical solution than infrastructure based approach. It eliminates the infrastructural and maintenance overheads as it makes the sensor network easier to evolve as the new sensor enabled entities appear in the environment. In addition, the design intends to isolate context gathering from context-aware applications. This would allow application developers to focus on the applications without having to worry about the complexity of integrating new sensors and gathering sensor data from different type of sensors.

In essence, a proximity sensing system can be regarded as two primary elements: Client and sensor networks. In this, The Client can be a mobile device that is equipped with multiple sensing technologies and software routines to augment mobile devices with awareness of their environment. Sensor networks can be a collection of networks that are made up of exiting infrastructures and *ad hoc* sensor networks. These sensor networks are collectively referred to as Environmental Sensors in the rest of thesis. The next section proposes an architecture for context gathering.

3.4.1 Proximity-Sensitive System Architecture Overview

This architecture consists of two main elements: Environmental Sensors and Clients (mobile devices), and is referred as Proximity-Sensitive System Architecture in this thesis. Environmental Sensors and Clients are described in detail below.

3.4.2 Environmental Sensors

In the proposed architecture, Environmental Sensors provide three different networks for gathering context information. They enable users' mobile devices to find themselves in relation to the entities around them. The networks include GPS, Bluetooth and RFID for supporting outdoor long-range, medium-range and short-range sensing respectively. The GPS network is provided through low orbit satellites to enable mobile devices find their position in the environment. The Bluetooth network is made up of a variety of mobile and static entities that are Bluetooth enabled (e.g. mobile phones, printers, ear phones). The RFID network is formed using passive RFID tags stuck on to informationally-interesting mobile and static entities. Together they form a flexible multi-sensor network for proximity sensing.

3.4.3 Clients: Sensor Enabled Mobile Devices

Mobile phones, PDAs and Smartphones were designed to support computing and communication on the move. The design proposed in this thesis takes advantage of these commodity items and uses them as Clients (mobile devices) in the Proximity-Sensitive System Architecture. These mobile devices are generally equipped with different types of sensors or are extendable to include sensors for providing the connectivity necessary for communication and other forms information access. The design has made use of these sensors in mobile devices, referred to as *integrated sensors* in this thesis, to augment their mobile devices with an awareness of their environment that can be used to facilitate context-sensitive services.

The hardware components on a mobile device will require an RFID reader, a Bluetooth sensor and a GPS receiver. In addition to these hardware components, two software components are needed to take part in proximity sensing: these are named as Explorer and Linker. The Explorer is responsible for periodically gathering context information about a mobile device's environment. It uses the integrated RFID, Bluetooth and GPS sensors on the mobile device to sense the device's presence in relation to its surrounding environment (more specifically, in relation to entities with RFID tags, Bluetooth sensors nearby or in geographical coordinates). The Linker is designed to provide the connectivity between Proximity-Sensitive System and other context-aware system components.

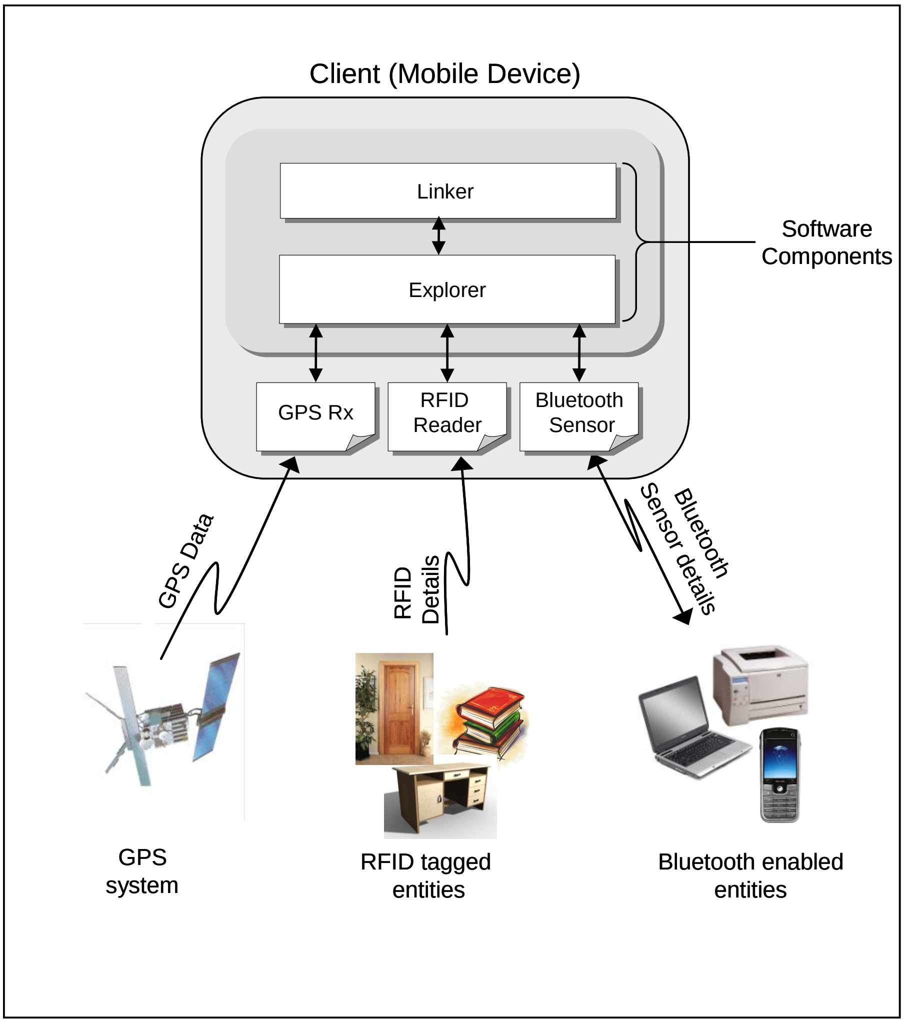

Figure 3.1: Proximity-Sensitive System Architecture

The Proximity-Sensitive System Architecture is illustrated in Figure 3.1. It shows a Client (a mobile device), Environmental Sensors (RFID tag and Bluetooth enabled entities and GPS). Within the mobile device, it shows RFID reader, Bluetooth sensor, a GPS receiver, and the two software components: the Explorer and Linker.

The next section explains the proximity sensing in more detail. It describes how the Explorer initiates the interaction between integrated sensors (Bluetooth sensor, RFID reader and GPS receivers) and their own type of Environmental Sensors (entities with Bluetooth sensors, RFID tags and GPS) to gather low-level sensor information. Whilst recognising that context is a complex and rich phenomenon, this low-level sensor information is referred as *context information* in the rest of the thesis.

3.4.4 Sensing between Client and Environmental Sensors

The Explorer running on the Client periodically initiates the discovery function for Bluetooth sensor, RFID reader and GPS receiver. Their operation is shown in Figure 3.2, and is described below.

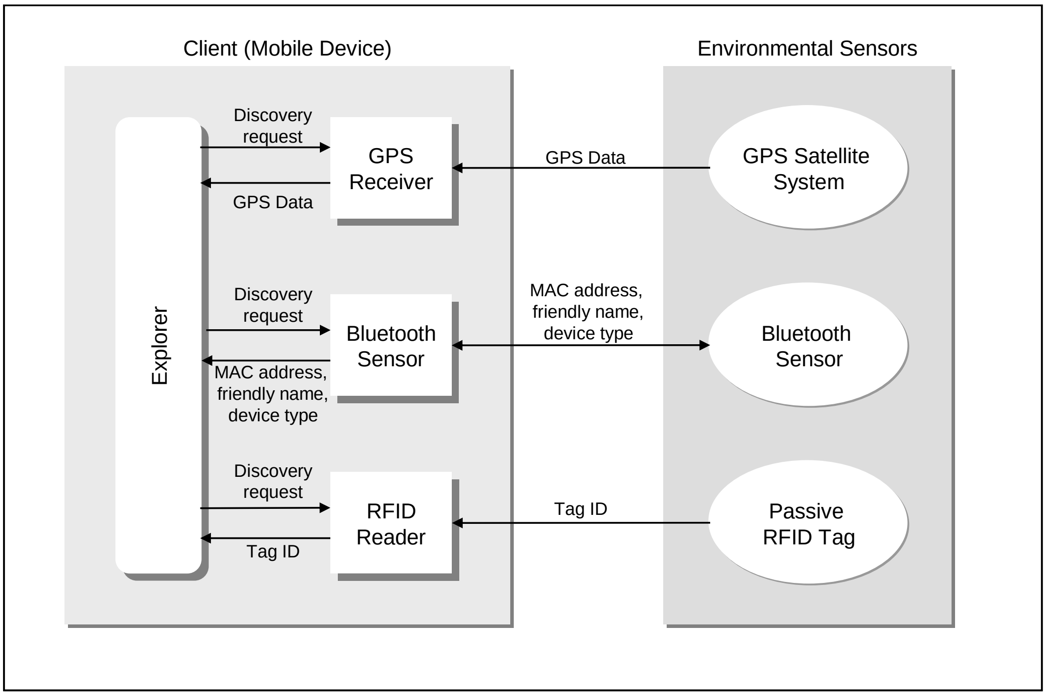

Figure 3.2: Information Flow between Environmental Sensors and Integrated Sensors

The discovery function for Bluetooth performs an inquiry, to find other Bluetooth enabled devices (environmental Bluetooth sensors) in the mobile device's current environment. Once Bluetooth enabled devices are found, the Explorer obtains their Bluetooth Medium Access Control (MAC) addresses, the devices' friendly names and device types through its mobile device's Bluetooth sensor. The MAC address uniquely identifies the Bluetooth sensor thus helps to identify the device it is attached to. The friendly name is often set up by the owner of the device to provide more information about the device. For example MAC address is made up of alphanumeric characters (e.g. 00:60:57:D4:98:50) which probably means nothing to most users. However, the friendly name is a user-generated name for the Bluetooth device and may be used to provide more useful information to senders or recipients (e.g. Ben's PDA). The device type offers information about the device itself (e.g. desktop printer), and is particularly useful to our design allowing users to make better informed judgements of their choice when associating messages to these devices, i.e. whether the message is related to static (a place or an object) or mobile (a person or an object) entity. Thus, if people can provide meaningful names for their devices (e.g. friendly name = Ben's PDA), then friendly name and MAC address may be sufficient to make better judgments about their choice of entities. For example, when the word PDA appears in a sensed device's friendly name, users can safely assume that this entity is most like to be mobile than static. Based on this reasoning, the decision was made to work with the MAC address and friendly name at this stage.

In similar way to Bluetooth, the discovery function for RFID gathers information about the RFID tag in the environment via the RFID receiver attached to the mobile device to indicate which entity is nearby. Like Bluetooth, RFID tags have unique identity numbers (referred as RFID tag ID) that help to uniquely identify themselves, and thereby the mobile and static entities that they are attached to. Nevertheless, these sensors typically offer little recoverable contextual information about their relationships to the entity they are attached to. For example, RFID tags are not normally labelled with information about their owner, or reason for its presence. However, due to the short-range RFID tag's characteristics (offering coverage up to a few centimetres), it offers support only when message senders are physically very close to tagged items, and so the relationships between the object and the sensor are likely to be visually evident to the sender (e.g. an RFID tag attached to Ben's desktop computer is likely to be Ben's or at least Ben is likely to have a relationship to that entity). Senders may therefore need to manage their message content to allow their recipients to make appropriate connections between the message and the entities or places that the message relates to.

What is interesting about Bluetooth and RFID is that the information gathered by the Explorer are already available to mobile device users (e.g. a Nokia 6600 phone with an active Bluetooth sensor is already broadcasting its MAC address, friendly name and device type to other Bluetooth sensors around it) and this information is simply utilised to augment mobile devices with awareness of their environment, i.e. identifying which entities are in close proximity to the mobile device.

In contrast to the above two technologies, GPS offers absolute location rather than relative (i.e. mobile devices are identified in relation to another entity in its environment). In the Proximity-Sensitive System Architecture, the GPS discovery function finds latitude and longitude using the data received by the GPS receiver. This information is then processed and converted into National Grid Reference (NGR) to locate the mobile device on a geographical map. GPS is incorporated into the system to provide proximity sensing in outdoor environment in relation to places and geographic locations. Like RFID, GPS also provides abstract values (i.e. NGR) rather than descriptions about entities. But this is less of a problem considering that a message sender has to be physically present in a geographical location to be able to send a message relating to an area around his or her current location. For example, a sender may want the message to be delivered when the recipient is within 100m radius from the sender's NGR: they cannot do this without being physically present at the NGR when sending a message. This allows senders to be aware of their locality (i.e. they can see where they are). Alternatively, a geographical map could be used to assist senders, showing their exact position on the map. The map could be even used to define an area for message delivery.

Another useful characteristic of GPS is that the NGR can be used to deliver messages at different levels of spatial proximity (e.g. 100m, 500m and 1km areas from NGR). For example, if a mobile device can identify its NGR using its GPS receiver, then use the NGR as a centre, and define an area (based on radius) of almost any size, this would allow message deliveries to take place at any level of proximal specificity.

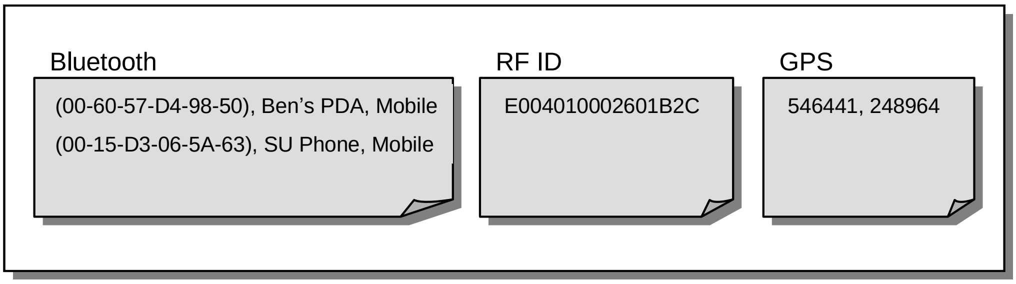

Figure 3.3: File Structure for Bluetooth, RFID and GPS Context Files

Finally, the Explorer stores the context information gathered by discovery functions locally on the mobile device, in their own *context files*. These *context files* (Bluetooth file, RFID file and GPS file) and the data stored in those files are shown in Figure 3.3. The Bluetooth file contains MAC addresses, friendly names and device types for the Bluetooth sensors discovered by the mobile device. The RFID and GPS files display the RFID tag ID for RFID tag discovered and the geographical coordinate for the mobile device respectively. The context information for Bluetooth, RFID and GPS are kept in separate files to provide flexibility and maintainability to systems. For example format or structural changes made to Bluetooth file will not have any impact on RFID or GPS files (i.e. their context files and discovery functions). The information stored in these files is made available to the user via the Messenger which is explained in the next Chapter.

The next section describes a generalisation of the technical Proximity-Sensitive System Architecture described here. The generic architecture shows that Proximity-Sensitive System Architecture can support a wide range of current and future technologies.

3.5 Generic Proximity-Sensitive System Architecture

The Proximity-Sensitive System Architecture presented is not technology dependent. It is 'open' to adapt to any current or future set of sensing technologies that have a unique identification number or have the facility to provide location information. In this section, we explore the properties of different technologies and their communicative roles and affordances that they offer, and explain how the architecture adapts to other existing and future sensors.

As discussed in Section 3.4, the technologies selected (RFID, Bluetooth and GPS) offer similar potential for marking, the three sensors that the system utilises are very different in terms of their operation. Two of these sensors (RFID and Bluetooth) are 'environmental', in that they require two sensors to be in physical proximity to one another, whilst the other (GPS) utilises the signals received from the global satellite system and thus, does not require a third party device to be in proximal range. It is therefore location-based, though proximally triggered.

There are other types of sensors that could be added to the system with different constraints to those that have been discussed, some of which may carry additional constraints and offer new opportunities for developing proximity-sensitive services. Indeed, the wireless environment has an incredibly rich existing infrastructure of uniquely identifiable resources, and provides a ready-made platform for developing proximity-sensitive applications. Examples of this within an indoor environment include potentially integrating infra-red and ultrasound beacons (e.g. Randell *et al.* 2002) and another related technology, audio networking (Madhavapeddy *et al*., 2003), in which spaces and things can be easily and cheaply demarcated with inaudible audio signals running off commonly available sound hardware, and which can be picked up with microphones on mobile devices. These technologies offer room-level precision, as infra-red and audio signals are bounded by walls, and this may be a useful property for controlling the range of the broadcast signal. As these are not already commonly available in the environment, this infrastructure would need to be created; however, this is not to say that such devices would not necessarily become more main stream at some later date.

WLAN and Barcode technologies are also worth exploring. WLAN has been seen in a number of existing location-based projects as noted earlier - generally WLANs have a greater range than the widely available Bluetooth. WLAN devices are less likely to be mobile (and therefore acting as a personal signifier) and like ultrasound beacons and audio networking will typically have less metadata with which to make interpretations about the context of the message (or proximity-related service) discovered. Even barcode offers opportunities for marking objects, and has the advantage of being extremely cheap to place. Indeed, like the barcode, any uniquely identifiable and easily captured information media can be used for marking, ranging from written and manually entered telephone numbers to visual tags (e.g. Madhavapeddy *et al.*, 2004), icons or pictorial images that can be automatically recognised by the camera in a mobile telephone. Where electrical power is available, it may be possible to physically mark environmental sensors to either support precise sensor discovery (e.g. determining what messages relate to), or simply to determine that a message has been associated with a sensor, and that it should be investigated further (e.g. by scanning a passive RFID tag).

Below, we are interested in exploring the characteristics of our Proximity-Sensitive System Architecture to describe how this architecture adapts to any current or future sensors. This can be best examined through looking more closely at the proximity-sensing process. Figure 3.4 shows the information flow between environmental sensors and integrated sensors in a generic Proximity-Sensitive System Architecture.

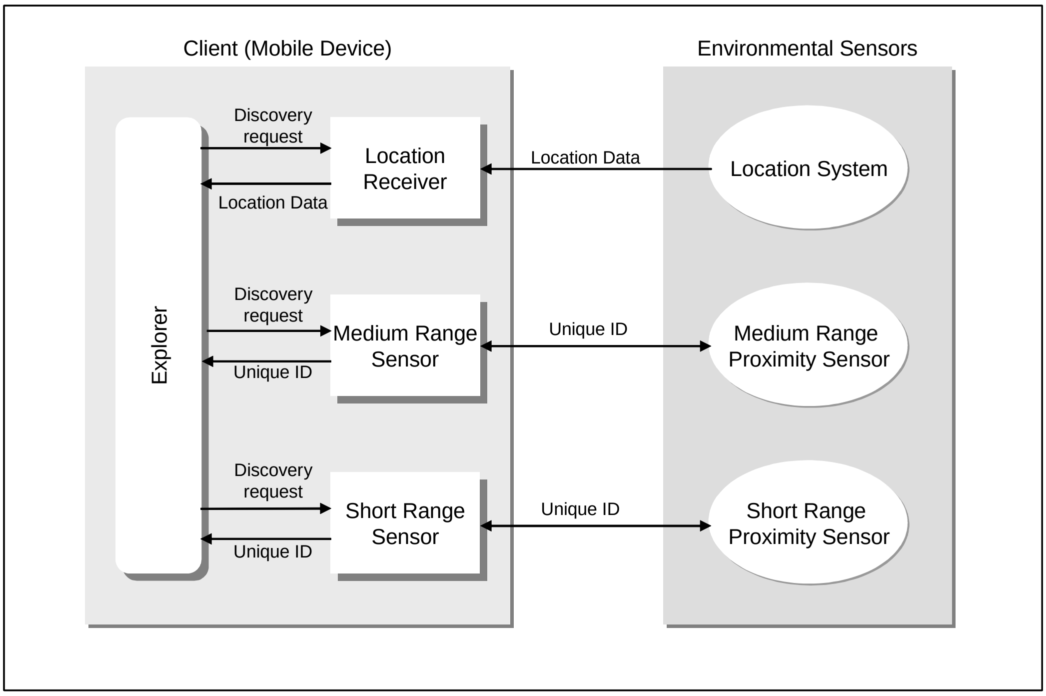

Figure 3.4: Information Flow in Generic Proximity-Sensitive System

The Explorer controls three different types of sensors in its user's mobile device to discover and mark different levels of proximity. These sensors offer short, medium and long range coverage and are shown within Client (mobile device). Sensors attached to people, places and objects in the environment are distinguished from the Client marked in the diagram as Environmental Sensors. The Explorer controls short, medium and long range Integrated Sensors and ensures that they discover and communicate with any short, medium and long range environmental sensors respectively. Explorer stores the context information (unique identity number of environmental sensor or location) gathered in appropriate *context files* (e.g. short.txt, medium.txt and long.txt). These three *context files* can be used by context-aware systems to provide a variety of context-aware services.

3.6 Summary and Conclusion

The chapter presents a set of requirements for supporting context-awareness in mobile and dynamic environments. Following on from these requirements, the chapter then describes that proximity offers a rich and distinctive set of characteristics compared to those of location and argues that its potential benefits extend further than simply providing awareness to users when an entity comes into proximity, explains its use in maximising coverage, supporting mobility and scalability. This chapter provides a review for a number of available sensing technologies, and discusses their suitability in meeting the system requirements. Finally, it introduces a Proximity-Sensitive System Architecture, its architectural elements and describes how the architecture helps to gather context information necessary for enabling proximity-sensitive services in mobile and dynamic environments. It also describes how the Proximity-Sensitive System is designed to adapt to various existing and future sensors. The next chapter examines how this context information is utilised in a proximity-sensitive personal messaging application through a Service Architecture.

Chapter 4: Supporting Context-Aware Services in Mobile and Dynamic Environments

4.1 Introduction and Overview

Context-awareness is central to context-aware services as they help to support information delivery where they are expected to be most relevant. Chapter 3 presented a Proximity-Sensitive System Architecture that enables context-awareness in mobile and dynamic environments. This chapter explains and elaborates on how this architecture can be utilised to offer context-aware services on top of it.

The first part of this chapter describes the components of the Context-Aware Service Architecture and explains how the Proximity-Sensitive System Architecture discussed in the previous chapter fits into the overall Service Architecture. The Service Architecture not only includes this Proximity-Sensitive System Architecture but also various other components that are necessary for providing context-aware services to mobile device users. Although it is not the intention of this thesis to provide solutions for all the components of the Service Architecture, they are discussed in this chapter for the purpose of creating and explaining the proof-of-concept prototype, through which the architectural and technical features of the Proximity-Sensitive System design may be demonstrated and evaluated. The second part of the chapter examines the overall design and describes its role in supporting three very different messaging scenarios.

4.2 A Context-Aware Service Architecture

So far, the thesis has focused on presenting an architectural solution for obtaining context information necessary for enabling context-aware services in mobile and dynamic environments. This section describes how the Proximity-Sensitive System Architecture fits into the Context-Aware Service Architecture (referred to hereafter as the Service Architecture). More specifically, it shows how the context information gathered by the Proximity-Sensitive System is utilised by the Service Architecture to enable context-aware services such as proximity-sensitive messaging services. It discusses functionality and design of the three main components: Proximity-Sensing, Interface and Routing necessary for supporting proximity-sensitive messaging services. It describes how these three components can be designed to interconnect and interact with each other to support proximity-sensitive messaging services.

Figure 4.1, illustrates the Service Architecture, highlighting the Interface and Routing components in blue to show where the Proximity-Sensitive System fits into the broader Service Architecture. It also shows how the Service Architecture interacts with the Proximity-Sensitive System through the Linker. Note that the Linker has no direct interaction (shown by discontinued arrows) with the Explorer or Routing component. These components are kept separate to provide flexibility to application developers. For example, changing the technology or method of message routing will not have an impact on other components of the Service Architecture. Below (Sections 4.2.1 to 4.2.3), these three components are described in detail along with their design options and design rationale.

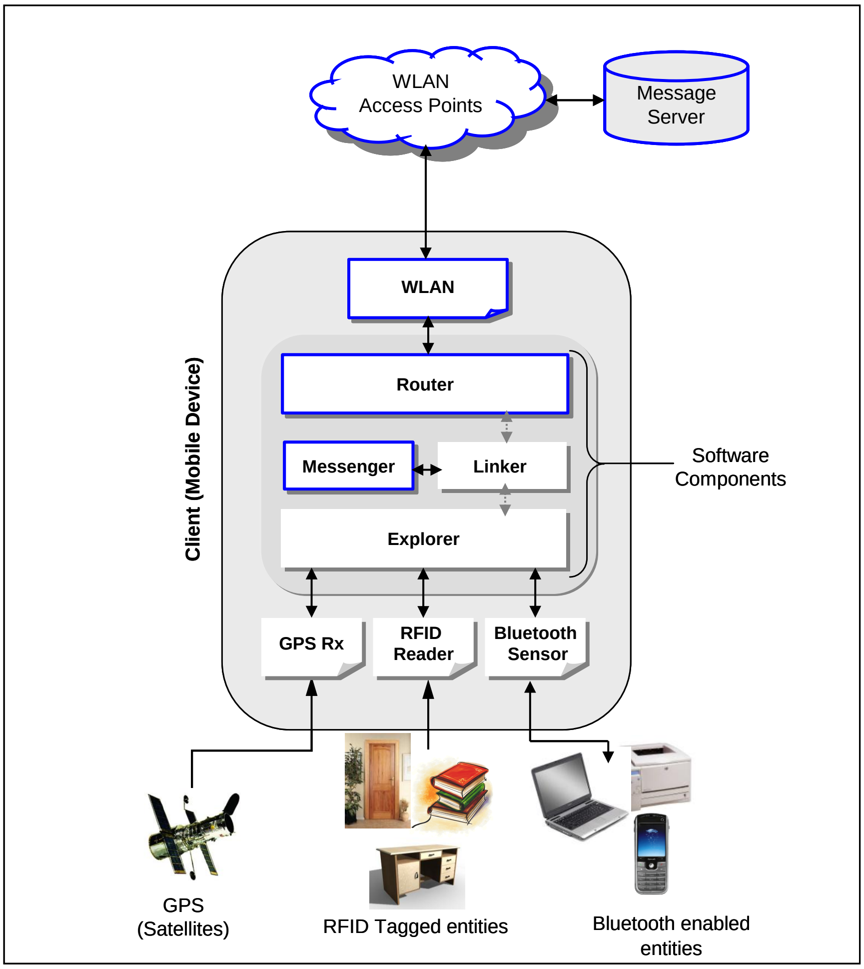

Figure 4.1: Context-Aware Service Architecture

4.2.1 Proximity-Sensing Component: Proximity-Sensitive System

Chapter 3 explains that the Proximity-Sensitive System is used to periodically perform proximity sensing (i.e. a form of context gathering) and direct the information into *context files*. In order to enable proximity-sensitive services, the information in *context files* has to be made available to the other components of the Service Architecture. This interconnection functionality is supported by the Linker, a software element in the Proximity-Sensitive System Architecture.

The Linker is a software program that is stored locally on mobile devices; it periodically gathers information from *context files* (see Section 3.4.4), and passes that information to the Interface component on the same mobile device. Additionally, it scans through a list of message files waiting to be delivered to its recipient on his or her mobile device, selects the ones that are relevant to recipient's current context, and makes them available to the Interface component. The Linker does not have direct interaction with the Explorer or Router (in practice, the software element of the Routing component). It uses *context files* created by the Explorer and message files downloaded by the Router to provide information to the Interface component, i.e. only the messages that are tagged to the information in the *context files* are sent to the Interface component.

4.2.2 Interface Component: Messenger

The Interface component discussed here is used for visually demonstrating the functionality of the system relating to the proximity and message-related information that is available from Proximity-Sensitive Services. Providing a multimedia support or creating an interface for usability testing is outside the scope of this thesis. Thus, it needs to be recognised that this interface design has not undergone usability testing and is not intended for end users. Moreover, the interface shown does not include the full range of functionality (e.g. multimedia) that can be supported by the architecture. Figure 4.2 presents the current version of the interface. The interface shows both send and receive functionalities.

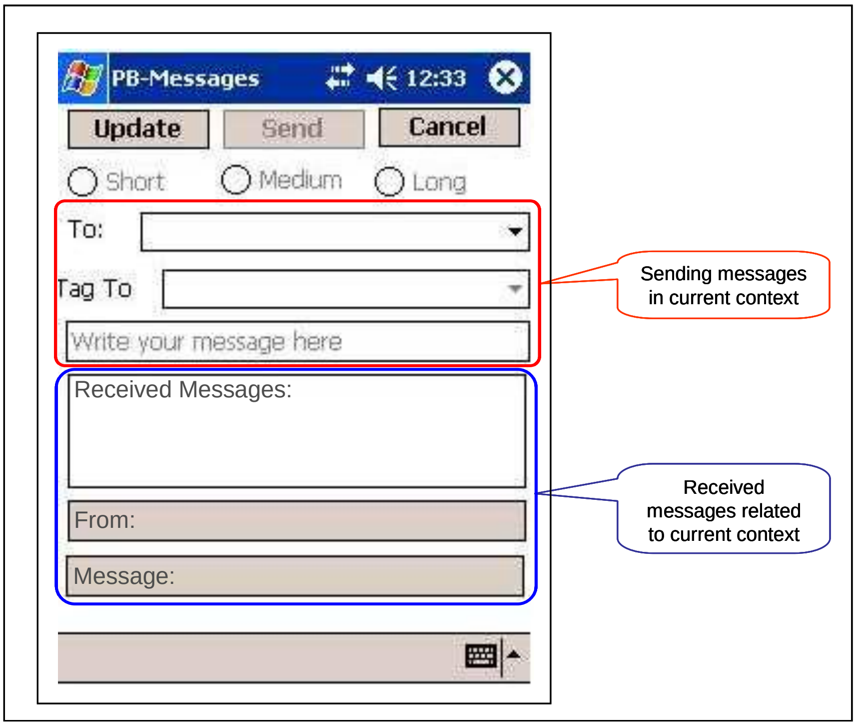

Figure 4.2: Interface Design

The Interface component is supported by a software application called the Messenger. The Messenger is installed on mobile devices to offer an interface for sending and accessing messages based on the information provided by the Proximity-Sensitive System. The context information (e.g. Bluetooth MAC address) is periodically acquired by the Explorer and is made available to the Messenger through the Linker. The Messenger then makes this context information available to mobile devices users.

4.2.2.1 Sending Messages

As can be seen, Figure 4.2 shows three radio buttons at the top of the interface that become activated (i.e. not greyed out) according to the entities available: thus if a GPS signal is available, 'Long' will be enabled, if another Bluetooth device(s) is discovered, 'Medium' will be enabled, and if an RFID tag is detected, 'Short' will be enabled. The sender can choose to 'attach' a message to an appropriate environmental entity using these radio buttons. If a person wishes to send a message, he or she will have to select an enabled radio button, then touch the

'To:' menu to choose a contact name to send the message. This will help to uniquely identify the message recipient (through his or her mobile phone number, e-mail address or URL), to forward the message. The 'Tag To:' input box (technically a combo box) is a dynamic menu that provides a list of discovered sensors or target ranges for current GPS location. This is dependent on the choice of range/sensor type that users have selected from the radio buttons. Thus, if a person has chosen to deliver a message within medium-range, it will display all of the Bluetooth sensors available, and will include both its MAC address and 'friendly' name; short-range will display RFID tag ID; and GPS will display three levels of Radius (100m, 500m 1 Km) for the current NGR. The text box below this ('Write your message here') allows a person to enter their message text.

Once the message is created and the Send button is pressed on the interface, the Messenger creates a message file. The Messenger uses recipient's details and context condition (contents of "To" + "Tag To") to generate the file name for the message file. It then writes the message content and sender's details to this newly created message file before saving the file on the sender's device in the /tosend/ folder.

4.2.2.2 Receiving Messages

Proximity-Sensitive messages are only made available to users when their device finds the environmental entity that is associated with their messages (i.e. marked entity). In Figure 4.2, the first box in the received messages part, lists all received messages that relate to the current context. This box displays the sender's name, alongside details of the sensors that have been discovered in the surrounding environment with their identifying details. The recipient can highlight a message from the received message list and see further details of this in the next two boxes: the first box displays sender details, while the second box shows the content of the message.

4.2.3 Routing Component

Once the messages are created by the sender, they have to be routed to recipients' devices, and there are many ways in which this could be carried out. These are discussed briefly before choosing a method for prototyping. The focus here is to create a quick solution for system demonstration rather than optimise for performance.

4.2.3.1 Message Storage Strategy

The Proximity-Sensitive System Architecture supports communication between two devices in relation to an entity in their surrounding environment. This is different to peer-to-peer information delivery, in which information from one device (a sender's device) is directly passed on to another device (e.g. bluedating), and thus cannot be supported without storing information on a remote third party device. Two options are considered here for storing information: storing locally on entities and storing remotely on a central-server. With few exceptions (e.g. Beale, 2005; Davis & Karahalis, 2005), almost all context-aware systems have used remote storage. They have not stored messages at the location or on entities, rather they have associated messages with relevant context (location or sensor) and stored them on a central information server, such as e-mail servers (e.g. Andronikos *et al.*, 2004) or SMS server (e.g. Mitchell *et al*., 2006). In order to store messages locally (on an entity), the Routing component would have to allow a mobile device user to store messages in another user's device. This would raise serious security and privacy concerns among mobile users. Further, entities such as books and doors will have to be equipped with storage space, making them to play an active part in messaging and run our proprietary software. For these reasons it was decided that the Service Architecture will follow the example of GeoNotes, comMotion, Social Serendipity and CoolTown, and utilise a central-server for storing messages.

There are many different ways in which central servers can be designed for prototyping: e-mail, SMS, http (Hypertext Transfer Protocol) or mobility servers.

An Http Server seemed more suitable for rapid prototyping when transferring small files and seemed a better choice for demonstrating the architecture discussed in this thesis without getting involved in writing complex programs for e-mail and SMS. In addition, it helps to avoid the financial overheads of mobility servers. For these reasons an http client server was adopted for this prototype design.

4.2.3.2 Connectivity Strategy

As messages will be stored on a central-server, the design needs to allow both the sender's and recipient's devices to connect to this central-server. There are two main technologies that offer wireless network connections to mobile devices: mobile phone networks (such as GSM, GPRS and UMTS) and WLAN. These networking technologies are already pervasive enough (LaMarca *et al*., 2005) to provide the coverage necessary to obtain a connection to a central server. The design was left 'open' to adapt to both technologies for future developments, although WLAN is used for prototyping purposes in the current design instantiation.

4.2.3.3 Routing Strategy

Once a connection is available, the messages have to be routed to and from the central server. A software application called the Router (see Figure 4.1) connects to the http server using a wireless networking technology (in our prototype, WLAN) available on the mobile device. This issues an http request and waits for its response; if the connection is successful, messages in the /tosend/ folder (i.e. messages waiting to be sent) are transferred to the message server. During this connection, messages waiting for this particular mobile user are downloaded from the message server to his or her mobile device. The downloaded messages are stored under /toread/ folder in recipient's device. However, there are different ways in which the downloading can be initiated. What follows lists the different ways in which messages can be delivered to recipients' devices along with rationale for the choice.

Downloading Options:

1. Periodically pull all messages that are relevant to the mobile device (recipient's device), where they are stored until the context condition is met.
2. Pull all entities (context information) that have messages to the mobile device, download messages on encountering those entities.
3. Send all entity encounters to the server to find out whether there are any messages for those encountered entities so that the messages can then be downloaded.

The download function can adopt any one of the above options to control the message delivery to the device. This download is only responsible for delivering relevant messages to the device and storing them in the device's local directory until the message notification is triggered. As far as the recipient is concerned, the message is not available for viewing (i.e. the recipient is not notified of the message) until the context condition is met. For reasons of optimising connection to the remote messaging server, the first option was chosen for prototyping. Consequently, the messages are downloaded and stored locally on recipients' devices until their devices discover a marked sensor. Notably, this message caching is purely intended as a practical solution to the problems of network latency, cost and connection issues; given a faster network connection and better coverage, it would have been more appropriate to use a real-time, proximity triggered message delivery system, as this would have a number of advantages, including problems arising as the number of potentially large size multimedia messages scaled up with their attendant memory demands on the mobile device. When the message recipient's device discovers a marked sensor, and they are identified as its message pointer, message delivery is automatically triggered. At this point the message becomes accessible to the Messenger through the Linker. The Linker makes the context information available to the Messenger, and Messenger makes the message available to the recipient when his or her device encounters the marked entity. In addition, the first technique provides a comparatively easy solution for prototyping the design as it does not require additional code for searching through message files for context information.

In summary, central server option was chosen for storing messages, and WLAN was chosen for providing the connectivity between the message server and mobile devices. A decision was also made to periodically pull all messages to recipients' devices (irrespective of meeting a context condition) and wait for them to encounter the marked entities before making them available for recipients to read at the user interface.

The Service Architecture presented here has a diverse set of possibilities in terms of designing individual components. For example, all the software components could have been built into a single application that was responsible for interface, routing and context gathering. However, the prototype design discussed here has kept these individual components separate to provide flexibility for future developers. In this way, future developers are free to choose technologies, development tools and techniques that are most suitable for their choice of services without impacting on their ability to gather context i.e. changing the Router will not have any impact on the Explorer.

4.3 Supporting Context-Aware Messaging

This section examines the Proximity-Sensitive Service Architecture and describes its role in supporting the two proximity-sensitive messaging scenarios discussed in Chapter 1. Example 1 below is used to describe how the design provides support for infrastructural and location independence discussed in Scenario 1. Examples 2 and 3 help to highlight the importance of different spatial specificity discussed in Scenario 2 and describe how these different granularities are supported by the design. For example, Andy leaving a message close to a motorway junction or in a building is very different to leaving a message on Ben's desk. Andy may want to choose long-range (more than 10m) for delivering messages in relation to a building and short-range (few centimetres) for delivering messages in relation to Ben's desk. This section highlights that each of these scenarios has a distinctive set of requirements (operating environment, coverage range and entity type), and describes how the architecture utilises technological convergence to adapt to their individual needs.

4.3.1 Example 1: Infrastructural and Location Independence

The first messaging scenario in Chapter 1 highlights the importance of mobility in context-aware messaging. The main challenge in designing systems to support this kind of messaging is that the interacting devices (message senders and recipients) and the entities (entities that are associated with messages) with which they interact are all mobile. The section below uses the scenario discussed in Chapter 1 as an example to describe how the Proximity-Sensitive System Architecture provides support for mobility through the use of Bluetooth sensors.

In the proposed Proximity-Sensitive System design, Bluetooth supports mediated (i.e. not directly peer-to-peer), medium-range messaging between people: it allows the sender to write a message, associate the message to an environmental entity with a Bluetooth sensor (discovered by the sender's mobile), and send it to recipients. In mediated messaging, there is a third party entity (an Environmental Sensor) that is associated with the message. In the first example scenario, Andy is the sender and Cathy is the message recipient. Ben's device (more specifically, the Bluetooth sensor attached to his device) is the Environmental Sensor which coexisted first with Andy's device, and then with Cathy's device. All entities involved in this messaging are mobile, taking the communication beyond places or static objects.

Figures 4.3 and 4.4, and the discussion that follows illustrate how Ben's Bluetooth sensor (attached to a mobile entity, e.g. a mobile phone) is used to support mobility; further, it provides details on the information flow between the entities involved in the messaging between Andy and Cathy's devices.

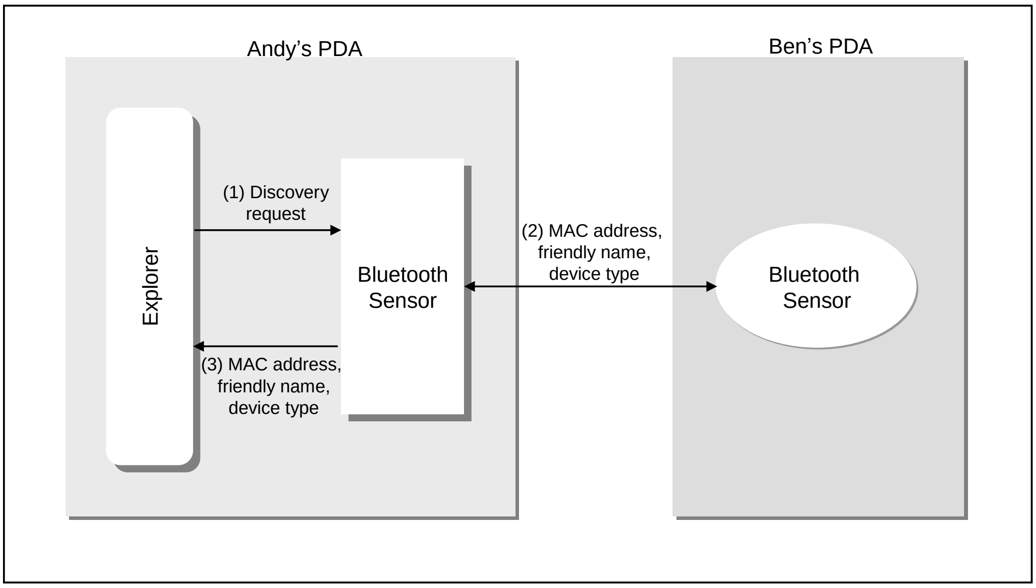

Figure 4.3: Information Flow between Andy's and Ben's Mobile Devices

In this instance, Ben's device does not require any special software because he does not actively take part in the messaging. However, both Andy's and Ben's Bluetooth sensors have to be turned on and Ben's sensor set to 'discoverable' mode. This will allow Andy's device to discover Ben's Bluetooth sensor and any other Bluetooth sensors that have come into proximity with Andy's Bluetooth device. Once the Explorer on Andy's device sets up the initial connection between his device and Ben's device, the exchange of information can take place between their devices. Bluetooth information flow takes place in both directions: Andy to Ben and Ben to Andy (see Figure 4.3). However, the Explorer on Andy's device only seeks to access Ben's Bluetooth MAC address, his device name (referred as the 'friendly name' within the Bluetooth standards) and device type. Whilst there may be connections and interactions between Andy's device and other coexisting Bluetooth sensors, they are not discussed here as they are not relevant to the message Andy is trying to send to Cathy.

Once the connection has been made, the Explorer on Andy's mobile device gathers Ben's Bluetooth MAC address, friendly name and device type, and updates its Bluetooth context file. The information stored in the context file is then made available to other elements of the Service Architecture in Andy's device. The Linker in the Service Architecture then makes the context information in this file available to Andy through the Messenger (i.e. the Interface); allowing Andy to create a message for Cathy and associate his message with Ben's device (i.e. to 'tag' the message against Ben's Bluetooth MAC Address). The message is then routed from Andy's device to Cathy's device via the wireless network.

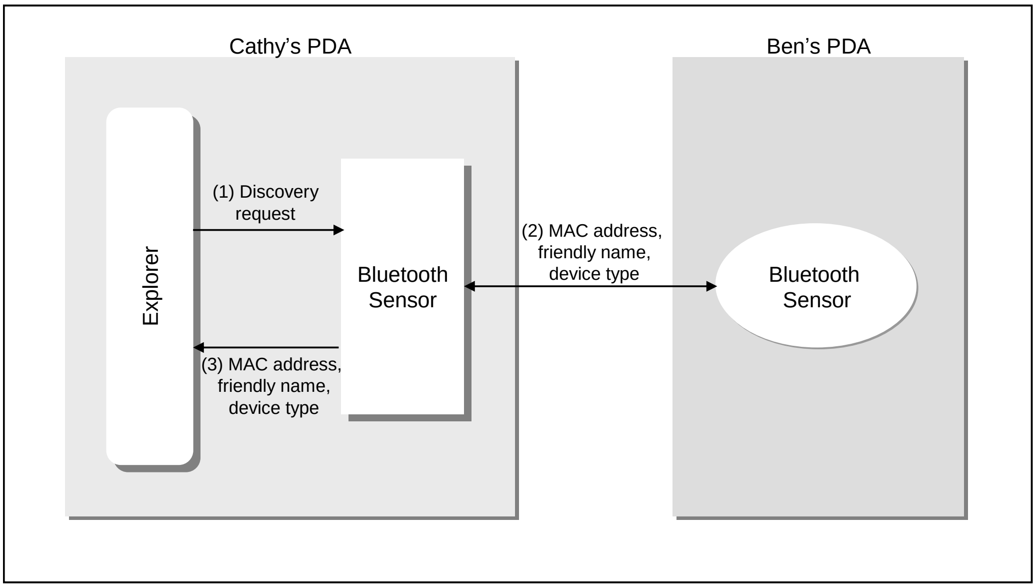

Figure 4.4: Information Flow between Cathy's and Ben's Mobile Devices

Like the Explorer on Andy's device, the Explorer on Cathy's device also periodically performs device discovery. When Ben's device coexists with Cathy's device, the Explorer in Cathy's device will find Ben's Bluetooth sensor. This will allow Cathy's Explorer to gather information on Ben's Bluetooth MAC Address, friendly name and device type (see Figure 4.4) in the same way as occurred with Andy's device. This information about Ben's Bluetooth sensor is then passed on to Cathy's Linker. This Linker then scans through the messages waiting for Cathy and picks up the messages that are associated with Ben's Bluetooth MAC address. The relevant messages are then displayed through the Messenger.

The above example scenario does not in any way suggest that Bluetooth technology is unsuitable for supporting messaging in relation to places or static objects. Indeed Bluetooth sensors can be attached to, or integrated within static entities such as printers, when the sender coexists with these devices the sender can leave a message for a recipient for delivery in this context (e.g. near the printer). As stated before (see Section 3.3.3), Bluetooth technology offers additional details that allow us to determine or at least make a reasonable guess about the entities discovered, such as whether they are static or mobile, their forms of use or the identity of their user/s. This information allows senders to decide whether they want to use Bluetooth to deliver messages in relation to these entities. For example, if the sender wants to deliver a message when the recipient is in a particular room, then the message must be tagged to an entity that *will remain* in that room. In addition, the sender must also know that the recipient is likely to coexist with this particular printer at some point.

4.3.2 Example 2: Indoor and Short-Range

The second messaging scenario in Chapter 1 discusses issues relating to spatial granularity, tailoring message deliveries to a few centimetres from the marked entity (i.e. the entity that is associated with the message). Additionally, it highlights an interest in associating messages with entities (mobile or static) that have no means of providing their own power. This section uses an example to explain how this situation is supported through the use of short-range RFID tags.

Consider a situation where Ben may have an interest in finding out whether a message has been left for him whilst he was out, and so may choose to stick an RFID tag to his desk. Thus, he will want to find out whether a message has been left on the tag when he is back at his desk. Alternatively, message senders may stick an RFID tag to Ben’s desk and associate their messages to it. Since it has been stuck to Ben’s desk, Ben is likely to have an interest in finding out what has been left on the RFID tag (in similar way to noticing a post-it note). Figure 4.5

and 4.6 shows the information flow between the entities involved in the messaging.

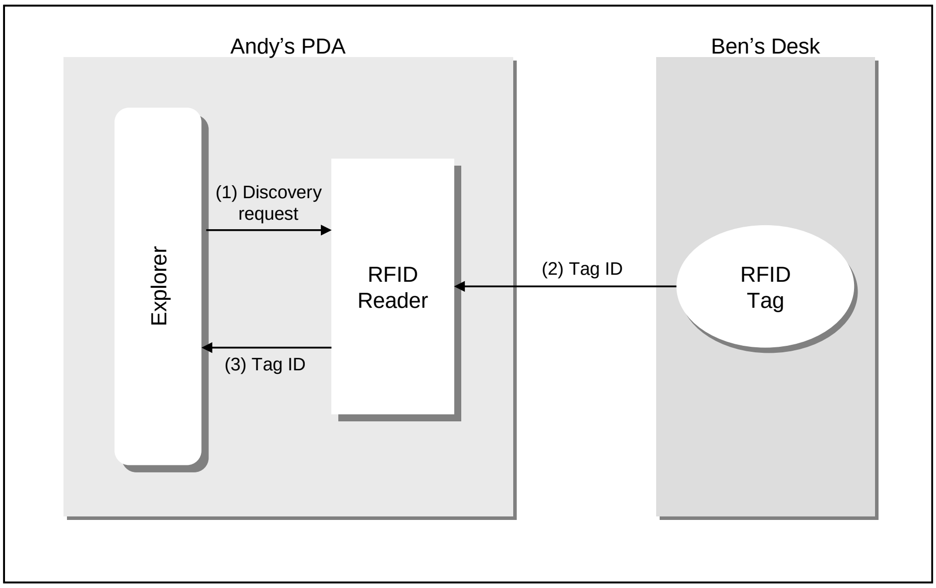

Figure 4.5: Information Flow between Ben's RFID Tag and Andy's PDA

The Explorer in Andy's device (specifically, Andy's RFID reader) finds the RFID tag left by Ben and retrieves its tag ID (Figure 4.5). This tag ID is stored in the RFID's context file on Andy's device. The information stored in the context file is then made available to other elements of the Service Architecture. The Linker in the Service Architecture then accesses this file and makes the RFID tag ID available to Andy though the Messenger on his device, allowing Andy to associate his message to this Tag ID. This message will then be routed to Ben via the wireless network from Andy's device.

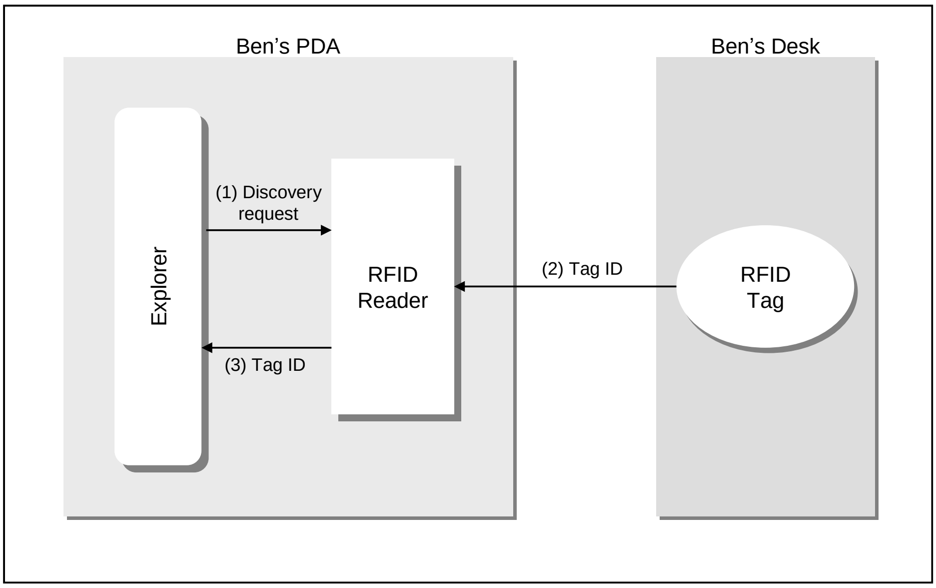

Figure 4.6: Information Flow between Ben's RFID Tag and Ben's PDA

Although Ben's device now holds the message, he can only view this message (see Figure 4.6) when his RFID reader finds the tag and retrieves the tag ID, by deliberate swiping or bring his device very close to the RFID tag on his desk. The Explorer in Ben's device gathers the tag ID using its RFID reader and makes it available to the other elements of the Service Architecture. The Linker in the Service Architecture then scans through the messages waiting for Ben and picks up messages that are associated with the tag ID left by him (or Andy). The message is then made available to Ben on his mobile device through the Messenger. In contrast to Bluetooth, RFID supports information flow in *only* one direction: from the passive Tag to the RFID reader. Unlike GPS, RFID passive tags can be used to support short range messaging in relation to static or mobile entities. For example, the RFID tag can be attached to Ben's book instead of the desk. In this case, Ben will get the message regardless of his book's location i.e. when Ben is in close proximity to his book.

4.3.3 Example 3: Outdoor and Long-Range

This section uses an example to highlight the importance of spatial granularity. More specifically, it explains how the design provides support for message delivery in large areas. Consider a situation in which Andy wants to deliver a message in relation to a particular motorway junction instead of Ben's desk, advising Ben to take an alternate route. Due to the changes in the operating environment and marked entity (i.e. the NGR of the motorway junction), this particular example demands a different set of requirements compared to those of the second example. Fine grained proximity sensing discussed in Section 4.3.2 is no longer useful. The section below describes how the GPS in the Proximity-Sensitive System Architecture provides support for this kind of messaging.

GPS offers context information that has been previously referred as 'absolute' location, and this absolute location is used by the proximity-sensitive services to find devices that are 'near' the marked NGR to support long-range outdoor messaging in relation to geographical locations. The Proximity-Sensitive System Architecture addresses scalability using two different techniques:

- It uses sensors with distinctive coverage capabilities (short-range RFID and medium range Bluetooth) to provide support for different levels of proximity scales.
- It uses GPS coordinates to define different size areas around geographical locations (i.e. to delimit area using radius).

To provide support for this particular situation, the second technique is used. GPS can in theory provide support for an area of any size. Figure 4.7 illustrates how different size (Radius R=100m, 500m and 1km) areas can be defined based on the NGR (X-Easting, Y-Northing).

Figure 4.7: GPS-based Scalability

To provide support for example 3, the Explorer on Andy's device uses the GPS receiver incorporated into his mobile device to gather GPS data. The GPS data are received as a sequence of strings and then converted into GPS coordinates by the Explorer before writing them to the GPS's context file on his mobile device. The information in the context file is then made available to the other elements of the Service Architecture. The Linker in the Service Architecture then accesses this file and makes the information available to the Messenger. The Messenger marks this NGR as Andy's current location. It then allows the message sender, (in this case Andy) to associate messages to an area around his current location. Figure 4.8 illustrates the data flow between Andy's device and the GPS system. The message is then routed to Ben using the wireless network on Andy's mobile device.

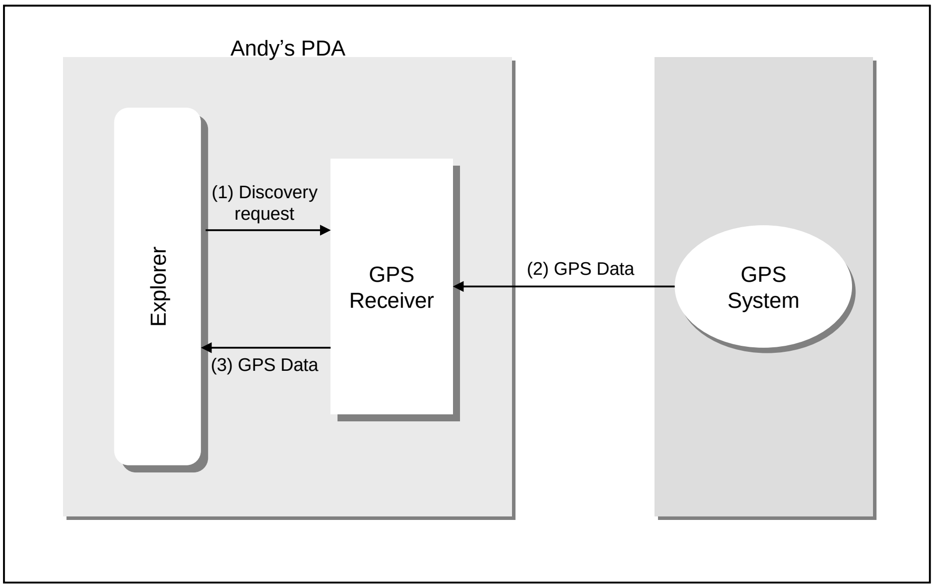

Figure 4.8: Data Flow between Andy's PDA and GPS System

The message is received by Ben's device and made available to him when he is within the area specified by Andy. For this to happen, Ben's Explorer must use the GPS receiver attached to Ben's device to gather GPS data, convert GPS data into coordinates, and then make his current location (NGR) available to the other elements of the Service Architecture. The Linker in the Service Architecture then scans through the available messages waiting for Ben and picks up messages that are relevant to his current location. Figure 4.9 illustrates the data flow between Ben's device and GPS system. The messages are then delivered to Ben on his mobile device through the Messenger.

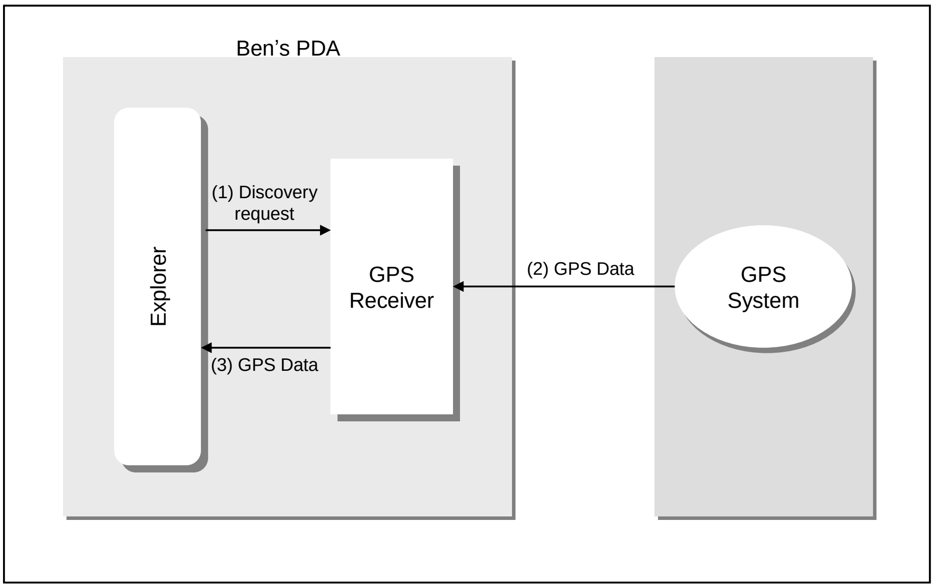

Figure 4.9: Data Flow between a Ben's PDA and GPS System

The three messaging examples discussed above help to exemplify how the proximity concept and convergence work together to support proximity-driven interactions across a variety of environments, across different levels of proximity and in relation to a wide range of entities.

4.4 Summary and Conclusion

This chapter introduced a Context-Aware Service Architecture and its components: Proximity-Sensing, Interface and Routing, for describing the creation of a proof-of-concept prototype. In addition, it provided details on where Proximity-Sensitive System Architecture fits into an overall Service Architecture, and its role in enabling proximity-driven interactions. Finally, it presented example scenarios to describe how the Service Architecture (through the use of Proximity-Sensitive System Architecture) exploits the proximity concept and unique characteristics of sensing technologies to enable scalable proximity-sensitive services in mobile and dynamic environments. However, the technology discussion in this chapter should not be seen as a solution for singling out technologies that are most suitable or optimised for Proximity-Sensitive System. It was presented with the intention to stimulate ideas relating to the proximity concept and building on technological convergence to support mobility and scalability. Furthermore, solutions proposed for Interface and Routing components have not been optimised for performance or usability testing and were essentially designed to demonstrate the potential of the Proximity-Sensitive System Architecture and for helping potential users to envision their use of such systems.

The next chapter describes the implementation aspects of the Service Architecture for providing scalable proximity-sensitive messaging services in highly mobile and dynamic environments.

Chapter 5: Proof-of-Concept Prototype for Proximity-Sensitive Messaging

5.1 Introduction and Overview

This chapter discusses the implementation of the Service Architecture for a Context-Aware prototype (see Chapter 4) that enables proximity-sensitive personal messaging on mobile devices. As stated, it is not the intention to offer a prototype solution for field trial, but rather, to provide a proof-of-concept that encapsulates an implementation of the ideas developed in the Proximity-Sensitive System Architecture. The chapter discusses the possibility of implementing the proof-of-concept using a set of devices, platforms and tools available for the research, within the many possible variations. At a practical level, this involves the implementation of the three components of the Service Architecture, namely, Proximity-Sensing, Interface and Routing. Although Interface and Routing are not the focus of this thesis, they require implementation to provide the user interaction and connectivity necessary to demonstrate the functionality of the Proximity-Sensing component.

The chapter begins with a description of the implementation of the Proximity-Sensing component and discusses how it interconnects with the Interface and Routing components to provide proximity-sensitive services (see Appendix A and B for implementation details on the Interface and Routing components). It continues with analysis of the implementation issues endured during the implementation of Proximity-Sensing component. Although the implementation issues identified are specific to the hardware and software used in the prototyping, they have broader relevance to mobile system development. These issues are important factors in *how* the implementation could be, and was, carried out in reality, and lessons can be learnt from this experience. For instance, when compared with desktop systems, mobile devices have limited resources in terms of processor power, storage space, battery power, screen size, etc., making it highly complex and difficult to develop applications for them. In addition, there are many variations and versions of mobile devices, operating systems and development tools, and this lack of standardisation widens the gap between the rhetoric and the reality of mobile application development. An application that has been tested on a development tool's device emulator often reacts very differently when it is deployed on a real mobile device that is interacting with a real wireless network.

The implementation aspect discussed in this chapter is relevant not only for the development of Context-Aware Services, but also, more broadly, to technological convergence and mobile application development. Furthermore, it demonstrates that the proposed design can be implemented using existing technologies without the need for a specialised infrastructure and custom-built hardware. The experience gained through the implementation process is used in this chapter to highlight the important factors that need to be considered when choosing the hardware and software for such systems.

5.2 Implementation of Proximity-Sensing Component

The Proximity-Sensing component consists of two parts: Environmental Sensors and Clients. The thesis, however, focuses on investigating how the existing sensor networks and infrastructures can be utilised to provide proximity-sensitive services. Thus, building a purpose built Environmental Sensor network was not necessary for prototyping. Appendix C shows some of the sensor enabled entities and infrastructures that could be used as Environmental Sensors for supporting proximity sensing.

Implementing the Client involves integrating hardware elements and developing the software elements (Explorer and Linker). The purpose of implementing the Client is two-fold. The first was to create a prototype for demonstrating the features of proximity. The second was to understand the challenges behind implementing such systems on commercially available and widely used mobile devices. The latter of these influenced the decision to utilise the hardware already available for the research. This included two PDAs: HP iPAQ h5550, with operating system Windows Mobile 2003 for Pocket PC, and HP iPAQ rx3715, with Windows Mobile 2003 for Pocket PC 2nd Edition. They both came with built-in Bluetooth sensors and Bluetooth stacks (i.e. baseband layer and software stacks that include a number of Bluetooth profiles), additional ports for supporting GPS and RFID, and built-in WLAN for wireless network connectivity. In addition to the mobile devices, a Compact Flash TeleType GPS receiver, a Bluetooth GPS receiver, a Compact Flash Socket RFID reader and a Phidgets USB RFID reader were also available for implementing the Client hardware.

Figure 5.1: Client Hardware

Figure 5.1 shows the hardware that is used in the implementation: a Bluetooth GPS receiver, a Compact Flash TeleType GPS receiver, an HP iPAQ h5550 and rx3715, with built-in Bluetooth and 802.11b, a Compact Flash Socket RFID reader with tags, and a Phidgets USB RFID reader with tags.

The implementation of the prototype also required a framework for developing the software elements on mobile platforms. Chapter 2 discussed the two main development frameworks available, namely .NET Compact Framework and J2ME, and provided information on their respective advantages and drawbacks. Although both J2ME and .NET Compact Framework have the potential to provide support for mobile application development, .NET was chosen for the following reasons.

- Both PDAs used for prototyping supported a Java Virtual Machine (JVM) called Jeode which was incapable of supporting J2ME. Jeode virtual machine was designed to support the ‘outdated’ Sun technology of Personal Java based on JDK. In addition, at the time of developing this prototype, there was no reference to a JVM that supported J2ME on the hardware and operating system of these PDAs that was available for free.
- .NET Compact Framework uses Windows Forms classes, including the full set of controls. Thus, it supports rapid development of user interfaces (MSDN Magazine, 2004).
- For J2ME, the Mobile Sensor API (JSR 256), was under development, and was not available for mobile devices at the time.

The Client software implementation was carried out using .NET Compact Framework and Microsoft Visual Studio C#. The next two sections provide insights into the implementation of the Client in two stages: the Client software Explorer and Linker.

5.3 The Client Software: Explorer

As explained in Chapter 3, the Explorer element is responsible for sensing a mobile device's current context and storing the context information in files. This is carried out through three technology-specific discovery functions namely Bluetooth, RFID and GPS, and context-update functions. Each technology-specific discovery function searches for its own type sensors in the mobile device's current environment or identifies the mobile device's location. Specifically, the Bluetooth discovery looks for other Bluetooth sensors within its coverage range (i.e. proximity range), the RFID discovery looks for RFID tag within its range and provides information about them, and the GPS discovery provides NGR for the device's current location. The technology specific, context-update function then saves the context information gathered by the discovery functions into technology specific files. Once the Bluetooth discovery is completed, the Bluetooth context-update stores the Bluetooth MAC address, friendly name and device type for all the Bluetooth sensors found in the current environment into the Bluetooth *context file* (BTcontext.txt). Similarly, the RFID context-update stores the RFID tag ID in the RFID *context file* (RFIDcontext.txt), and the GPS context-update stores the NGR in the GPS *context file* (GPScontext.txt).

The pseudocode for Explorer is given in Figure 5.2. This pseudocode helps to describe the underlying programming code for Explorer in structured English without going into the details of programming syntax. Each function in the pseudocode is named and denoted, and is referenced in the text. The pseudocode shows that the Explorer has separate discovery and context-update functions for Bluetooth (referred as *EB1* and *EB2* in the pseudocode), RFID (*ER1* and *ER2*) and GPS (*EG1* and *EG2*), respectively. The sections below provide details on Bluetooth, RFID and GPS discovery and update functions.

```
Explorer ()

   //Search for Bluetooth sensors and gather their info.
   Do Bluetooth discovery ()                    - (EB1)

   //write gathered information to Bluetooth context file
   Do Bluetooth context-update ()               - (EB2)

   // Scan for RFID tag gather RFID tag ID
   Do RFID discovery ()                         - (ER1)

   //Write RFID tag ID to RFID context file
   Do RFID context update ()                    - (ER2)

   //Receive data from GPS and convert to Easting, Northing
   Do GPS discovery ()                          - (EG1)

   // Write Easting, Northing to GPS context file
   Do GPS context-update ()                     - (EG2)
```

Figure 5.2: Explorer Functions

5.3.1 Bluetooth

The Bluetooth discovery function is responsible for finding other Bluetooth sensors within the coverage range of the mobile device and gathering information about those sensors (see Section 3.4.4). To do this, the mobile device needs an application called Bluetooth stack that enables devices to locate each other and establish a connection. Through this connection, Bluetooth discovery gathers MAC address, friendly name and device type relating to sensors discovered in the current environment. Without a Bluetooth stack, the Bluetooth sensor enabled devices will not be able to find each other to establish a connection and exchange information. Most Bluetooth enabled windows mobile devices come with either the Microsoft (Microsoft, 2008) or Widcomm (Broadcom, 2008) Bluetooth stack. However, these two stacks have very different Application Programming Interfaces (APIs), thus making it impossible for code written for a Widcomm stack to run on a Microsoft stack and *vice versa*. This incompatibility makes the Bluetooth implementation process problematic (see, for example Figueira, 2006) for developers in general, and for the prototyping in this thesis.

The HP h5550 and rx3715 devices chosen for prototyping were equipped with built-in Bluetooth sensors. Both these devices have the Widcomm Bluetooth stack and fail to support the Microsoft Bluetooth stack. In addition, the .NET Compact Framework v 1.0 and 2.0 were unable to provide native support to the Widcomm Bluetooth stack. Hence, Franson's BlueTools SDK (Franson, 2007) for Compact Framework version 2.0 was purchased to access the Widcomm stack on these devices. The Franson's BlueTools SDK provides access to the Bluetooth inquiry source code (written in C#) and allows incorporation of the code sections necessary for gathering MAC address, friendly name and device type for the discovered sensors. Figure 5.3 below provides the pseudocode for Bluetooth discovery. The pseudocode shows the tasks performed by the Bluetooth Discovery function. It shows that Client searches for other Bluetooth enabled devices in the environment (referred as *B1* in the pseudocode), gathers MAC address, friendly name, etc. and stores them in array (*B2*). In addition, it highlights that the discovery function removes discovered devices when they move outside the coverage range of the Client (*B3*).

```
Bluetooth Discovery ()                        - (EB1)

  //Start the inquiry - search
  Start Device Discovery (…………)               - (B1)
  {
    Search for Bluetooth devices
  }

  //Gather information when a new device is discovered
  Device is Discovered (…………)                 - (B2)
  {
    Store MAC address, friendly name and device type to
    BTdevice array
  }

 //Device is moved out and lost connection
 Device is Lost (……….)                         - (B3)
  {
    Remove device from the BTdevice array
  }

  //End discovery
  Device Discovery is Completed(…….)
  {
    Stop discovery
  }

 Return BTdevice array
```

Figure 5.3: Bluetooth Discovery

Once the Bluetooth inquiry initiated by the Bluetooth discovery function is completed, the Explorer starts the Bluetooth context-update function to write the Bluetooth information gathered during the inquiry process (contents of BTdevice array) into the Bluetooth context file. If the inquiry returns an empty array, that is, it found no sensors, then the Bluetooth context file will not have any entries, that is, the file will be empty.

5.3.2 RFID

In the case of RFID, the Explorer initiates RFID discovery and context-update. The RFID works in a similar fashion to Bluetooth discovery and gathers the RFID tag ID for an RFID tag nearby. The SocketScan RFID used in the prototyping comes with the SocketScan software that provides the RFID discovery function. In addition, this software helps to enter the discovered tag ID into an active windows application as virtual keystrokes. Thus, SocketScan provides support for both gathering context information (discovery function) and then writing the information into RFID’s *context file* (context-update), so obviating the need to write custom code for the RFID function.

5.3.3 GPS

GPS discovery works very differently to Bluetooth and RFID discoveries. It obtains latitude and longitude coordinates via a Compact Flash GPS receiver connected to the Client's serial port (i.e. iPAQ h5550), using the NMEA sentences protocol (Betke, 2001). NMEA sentence is a standard used by GPS receivers to transmit data. Once the latitude and longitude are extracted from the sentences they can be converted into Easting and Northing, and stored into the GPS *context file*.

Reading NMEA sentences through the serial port is the next step in GPS discovery. There are many different types of NMEA sentences and only a few are useful for determining the NGR of the Client. Each type of NMEA sentence is delimited by a return character and begins with a six character tag, the first character of which is a '$'. Although decoding these sentences is reasonably straightforward, reading from a serial port, managing the redundant and missing data, and then converting it to Easting and Northing is a tedious and time consuming process. To handle these details there are many third party tools such as Franson GPS Tools (Franson, 2006), GPS Toolkit (ScientificComponent, 2006), and GeoFrameworks (GeoFramework, 2008). For the purpose of reading GPS data from a serial port, the Franson GPS Tools is used. This data can then be converted and stored in the GPS *context file*. Figure 5.4 lists the pseudocode for Franson GPS discovery. It shows how the received NMEA sentences are split into words at the commas (referred as *G1* in the pseudocode), and how it obtains latitude and longitude from sentences if the first word of the sentence is "GPRMC" (*G2.1*). The same function then converts the latitude and longitude into Easting and Northing (*G2.2*).

```
GPS Discovery ()                                           - (EG1)

//Set ports to receive signals
Enable Comport()

//Receive GPS signals - sent in NMEA format
Receive NMEA sentences ()

//If GPS fix is achieved
On GPSfix
{
 //translate GPS data into meaningful data
 Split sentences into words at ','                          - (G1)
 {
   // sentences with prefix GPRMC is relevant
   If first word = "GPRMC"
   {
     Find positional info.(latitude,longitude)        - (G2.1)
     Convert positional info. into easting,northing  - (G2.2)
   }
 }
}

Return (easting, northing)
```

Figure 5.4: GPS Discovery

5.4 The Client Software: Linker

The Linker is responsible for two main functions (see Chapter 4): reading *context files* and passing context information to Messenger, and scanning files in the /toread/ folder to identify messages that are relevant to recipient's current context.

The first function in the Linker opens Bluetooth, RFID and GPS *context files*, one by one, in *read mode* and reads the Bluetooth data, RFID data and GPS data into the Bluetooth array, RFID array and GPS array, respectively. These arrays are then passed on to the Messenger on the Client through the Linker. Figure 5.5 details the sequence on how the data in Bluetooth, RFID and GPS *context files* are read into individual arrays.

Figure 5.5 shows that while there are entries in Bluetooth *context file*, Bluetooth data is read into a Bluetooth array (*LR1*). The Bluetooth array and Bluetooth count are then made available to other components of the Service Architecture. In a similar way, the Linker reads the context information in RFID (*LR2*) and GPS (*LR3*) *context files*. However, these two files will only have single line entries compared to Bluetooth's single or multiple lines depending on the number of sensors discovered by the latter (see Figure 3.3 for Bluetooth, RFID and GPS *context file* format), because multiple Bluetooth devices can be discovered simultaneously, unlike RFID and GPS. These three arrays are then made available to the second function in the Linker. The second function uses the information in the array to scan for messages that are relevant to the Client's current context.

```
Linker Read function ()

//Bluetooth
If Bluetooth context file is empty ()
{
   Bluetooth count = 0
}
Else
{
   While not end of file ()                    - (LR1)
   {
     Read a line at a time
     Split the data in each line at ',' into MAC address,
     Friendly name and device type
     Store MAC address and friendly name into Bluetooth-
     array
     Increment Bluetooth count variable by 1
   }
}
Endif


//RFID
If RFID context file is empty ()
{
   RFID count = 0
}
Else
{
   If file not empty ()                        - (LR2)
   {
     //RFID reader only discovers one tag at a time
     //therefore RFID file only contains one line
     Read the line, store data into RFID-array
    }
}
Endif

//GPS
If GPS context file is empty ()
{
   GPS count = 0
}
Else
{
   If file not empty ()                        - (LR3)
   {
     Read the line,
     Split the data in the line at ',' into Easting and
     Northing (NGR)
     Store NGR into GPS-array
   }
}
Endif
```

Figure 5.5: The Linker Reading Context Files

The Messenger creates message file names using the recipient's address and the context condition (see Section 4.2.2.1). The context condition format, however, varies with the range (Short, Medium or Long) chosen for messaging. Table 5.1 below provides details on context condition formats for RFID, Bluetooth and GPS.

Messaging Range	Technology	Context Condition Format
Short	RFID	RFID tag ID
Medium	Bluetooth	(MAC address) and Friendly name
Long	GPS	NGR and Radius

Table 5.1: Context Condition Format for Different Messaging Ranges

To illustrate, consider a messaging situation where a message needs to be delivered to Cathy when her Bluetooth sensor discovers Ben's Bluetooth sensor (attached to his mobile device). In this case, the message file will have Ben's Bluetooth MAC address and friendly name (in the format: (00-60-57-D4-98-50) Ben's PDA) in its filename. This file will be routed to Cathy's device and stored in her local /toread/ folder. The Linker on Cathy's device checks the context condition on the message file (that is, Ben's MAC address) against the current context information (stored in Bluetooth array). If a match is found (i.e. if Cathy's Bluetooth sensor has discovered Ben's Bluetooth sensor) then the message file is added to the For-Viewing list.

RFID based message delivery also works in similar way, that is, the RFID message file will have the RFID tag ID in its filename. The Linker on the recipient device compares the RFID tag ID in the *context file* with message filenames in the local /toread/folder. If a match is found then the message file is added to the For-Viewing list.

GPS based message delivery works very differently to Bluetooth and RFID message delivery. The GPS has an additional parameter, that is, the radius R (e.g. 100m, 500m, or 1Km), which is set by the sender at the time of message authoring. This R defines the message delivery area which is also referred to as messaging area. On a regular interval, d which is the distance between sender's NGR (x_s, y_s at the time of sending the message) and recipient's current NGR (x_r, y_r) needs to be calculated. This helps to determine whether the recipient's current location is within the messaging area. This calculation is carried out using a simple mathematical formula.

$$d = \sqrt{\left[(x_s - x_r)^2 + (y_s - y_r)^2 \right]}$$

If the value for distance d is less than or equal to the Radius R, then the recipient is in the messaging area (i.e. it meets the context condition), so message file is added to the For-Viewing list. Figure 5.6 below shows the sender's NGR and the messaging area with radius R defined by the message sender. It also shows that when distance d is less than R, then the recipient (current NGR) is within the messaging area.

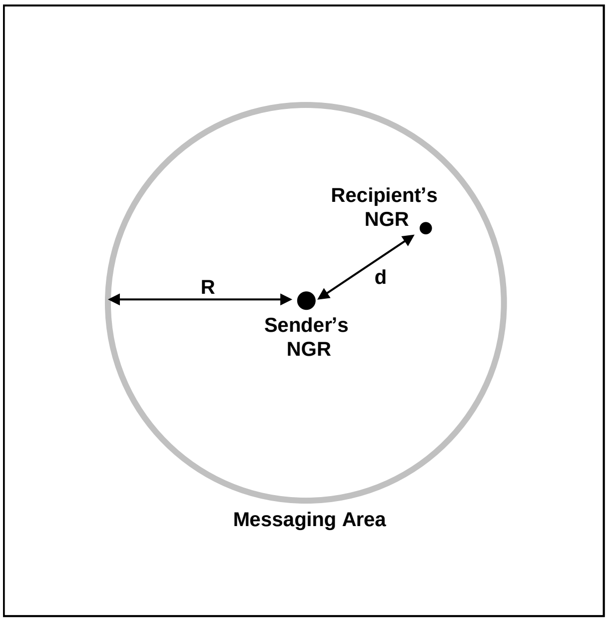

Figure 5.6: Messaging Area

Figure 5.7 provides details on the scan function in the Linker. It shows how each file in /toread/ folder is read and context condition extracted from its message name (see function denoted as *LS1* in Figure 5.7). These messages are then classified into RFID, Bluetooth and GPS respectively (*LS2*), before they are compared against the current contextual information gathered by the Explorer. If they match (if the message condition matches the content of current RFID (*LS3*), Bluetooth (*LS4*) array or if d <= R for GPS) – (*LS5*) the message file is added to the For-Viewing list.

```
Linker Scan function
For each file in /toread/ folder

 Split filename into recipient address and context condition
 Store context condition -> message-condition             - (LS1)

 classify Technology (RFID, Bluetooth or GPS)             - (LS2)

 Do case
  Case RFID
       If message-condition = current RFID tag ID         - (LS3)
       {
         Add the message file to the For-Viewing list
       }
  Case Bluetooth
     If message-condition = an element in Bluetooth array - (LS4)
     {
        Add the message file to the For-Viewing list
     }
  Case GPS
       Split contents of message condition into X, Y, R
       If (d<= R)                                         - (LS5)
       {
        Add the message file to For-Viewing list
       }
 Endcase
```

Figure 5.7: The Linker Scanning for Relevant Messages

Once the Linker has checked all the files in the /toread/ folder, the For-Viewing list is made available to the Messenger.

Linker provides the connectivity between the components in the Service Architecture. Figure 5.8 shows the data flow between the Linker and other components, how the context data gathered by the Explorer is made available to other components. It also illustrates that received and sent messages flow between Messenger and Router via Linker. Discontinued arrows between Explorer and

Linker, and Router and Linker indicate that there is no direct connectivity between these components. However, the information generated by Explorer (context information) and Router (messages) are processed by the Linker before passing context information and messages to relevant components.

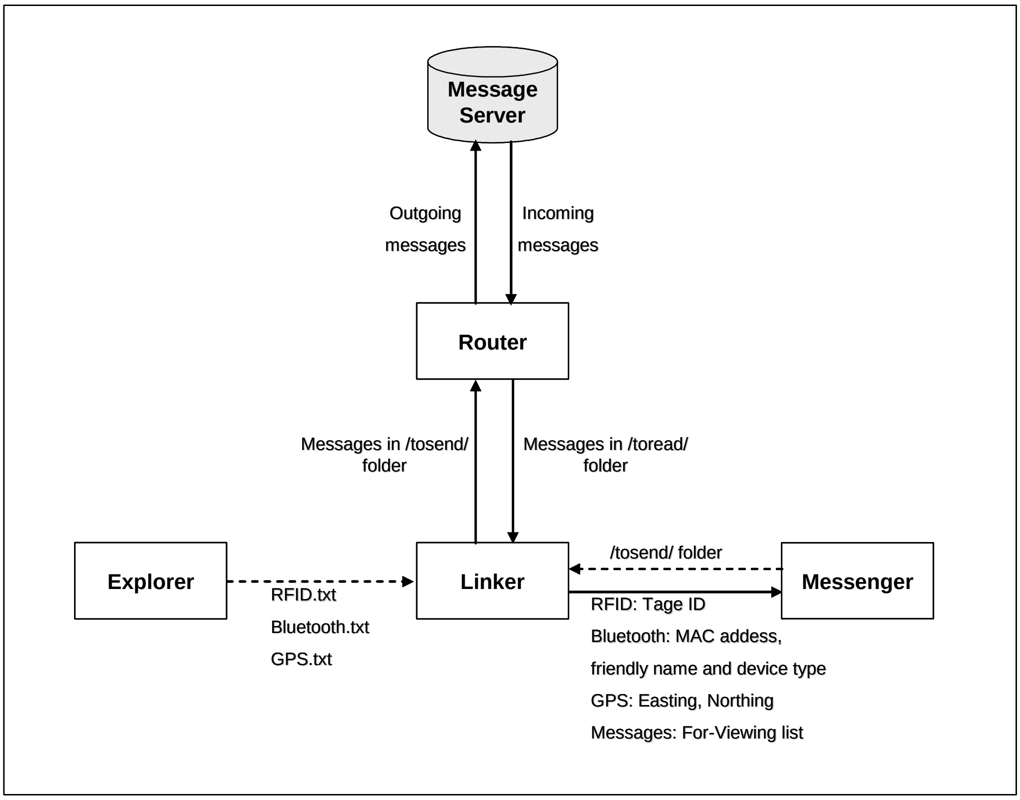

Figure 5.8: Data Flow between System Components

5.5 Implementation Summary

Table 5.2 summarises the implementation of Proximity-Sensitive Service Architecture. It shows various hardware configurations, hardware specific software and custom software used for each component of the Service Architecture.

Component	Hardware	Hardware Specific Software	Custom Software
Proximity-Sensing	RFID: CF Socket Scan and Phidget readers. Bluetooth: Built-in Bluetooth sensors GPS: TeleType and Bluetooth receivers	Socket Scan and Phidgets software drivers. Widcom stack Came with PDAs TeleType GPS driver	N/A Customised Franson's BlueTools Customised Franson's GPSTools and GPSGate.
Interface	N/A	N/A	C# code on .NET Compact Framework
Routing	WLAN hardware on PDAs and university server	Preinstalled WLAN software on PDAs	C# code on .NET Compact Framework

Table 5.2: Implementation Summary for Components

The Proximity-Sensing component implementation uses three different sensors for offering mobility, scalability and maximum coverage. RFID implementation was carried out using Socket RFID and Phidgets RFID readers, and manufacturer specific software. Bluetooth implementation utilised the Bluetooth hardware that came with HP iPAQ h5550 and HP iPAQ rx3715 together with Widcomm stack. Customised Bluetooth code was also created using C# version of Franson's BlueTools for gathering Bluetooth specific sensor information and directing that information to a text file (Bluetooth.txt). GPS element in Proximity-Sensing component was created using a Compact Flash TeleType GPS receiver, a Bluetooth GPS receiver and a customised version of Franson's GPSTools and GpsGate. Interface component in the Service Architecture did not require any hardware and its software element Messenger was implemented in C#. Routing component utilised the WLAN came with iPAQ h5550 and rx3715, and the university http server. In addition, software element Router was implemented in

C#. It must be noted that the implementation choices discussed in Chapter 5 is one of the many possible variations for implementing the Proximity-Sensing component discussed in the thesis.

5.6 Integration and Deployment

In the Proximity-Sensitive System, the static and mobile sensors located in the world make the Environmental Sensors. This provides a readymade infrastructure for proximity sensing. However, different types of sensors and various software components (Explorer, Linker, Router and Messenger) need to be integrated into mobile devices to create Clients.

Mobile devices and sensors come in different forms and configurations, thus integrating and deploying systems on mobile platform is rather challenging. The two Hewlett Packard PDAs chosen for prototyping came with pre-loaded Window Mobile 2003. However, they both behaved very differently when they were connect to a wide range of sensors and linked to software necessary for creating the proof-of-concept prototype. The proof-of-concept prototype uses RFID, Bluetooth and GPS for proximity sensing, and WLAN for connectivity, all of which have different port and software requirements.

Creating Clients brought different sensors with distinctive characteristics and different hardware, software and resource requirements into a single system. In contrast to a single sensor based system, this multi-sensor approach required additional ports, software and resources to incorporate various sensors. During the software integration phase of prototype development, the individual software components (Explorer, Linker, Messenger and Router) were brought together into a single application and built into a ProxMS (Proximity-Sensitive Messaging System) cabinet file, referred to as a "CAB" file, based on the file extension .CAB. There are different types of CAB files, each supporting a different type of target platform processors (StrongArm, MIPS, SH3 and X86). In addition to the processor, there are different variations and versions of mobile operating systems, and these are very different to the desktop operating system. These have some common features and share common functionality, but there are differences in the development approaches required for each. This is due to differences in screen sizes, resources and the APIs supported. It is therefore, necessary to build a .CAB file that can work on the target device type and its operating system. As both iPAQ h5550 and rx3715 used for prototyping have ARM processors it was possible to deploy the same CAB file on both devices using Active Sync version 4.2. In reality, a single build will not be sufficient to support all the mobile devices that are commercially available. This will require developers to create different builds for different devices.

5.7 Implementation Issues and Lessons Learnt

The development of the proof-of-concept prototype posed a number of challenges during the various stages of its implementation. The problems faced were not necessarily caused by hardware and software choices made for prototyping, but were more related to the issue of mobile application development that takes advantage of technological convergence. It is worth noting that the mobile devices and tools chosen were not only commercially available but they were advanced and in common use at the time of prototyping. However, the spectrum of available hardware ranges on mobile devices and the lack of standardisation found among those mobile devices make it difficult to incorporate multiple technologies into a single system. In contrast to desktop developers who deal with applications that target a single operating system, mobile developers often face situations where they have to adapt to different devices, operating systems and versions. In addition to the manufacturer specific operating systems, there are three main operating systems in the market. They are namely Windows Pocket PC, Windows Mobile and Symbian, and they all support different type of implementation (Zyda, 2007). For example, the PDAs come in different sizes, with different hardware configurations, operating systems, versions, development tools, connectivity choices, and APIs, and trying to develop a solution that could accommodate all these variations is problematic. As a result, even manufacturers of these devices or software vendors do not cover all variations in their documentation. Thus, mobile developers often find themselves looking for information that is not readily available or public knowledge when they encounter problems. In order to provide information to such developers, the issues faced during prototyping are described under the headings of Convergence Issues, Technology Specific Issues and Deployment Issues along with the situations in which they occurred, and their causes.

5.7.1 Convergence Issues

The Client in the Proximity-Sensitive System brings together multiple sensors into one mobile device. This convergence increases the power, memory and port requirements for the mobile device. In practice, using more than one sensor and wireless connectivity on an iPAQ h5550 not only drains the battery power rapidly but it also makes the application run slower. In addition, the port requirement for prototyping requires three different input and output ports on the device: one to connect with the GPS receiver, the other two to connect with the RFID reader and Bluetooth sensors, respectively. However, the device used for creating the proof-of-concept is an iPAQ h5550 Pocket PC with ARM processor which comes with a built-in Bluetooth sensor, leaving a requirement for two Compact Flash slots, one for a Compact Flash TeleType GPS receiver and the other for a Compact Flash Socket RFID reader card. The device, however, offered only a single Compact Flash slot. Although a USB Phidgets reader was available it was unsuitable (for details see next section), making it difficult to support proximity sensing at three different levels at any given time. Given the requirements for RFID to be manually selected, this does not pose a specific problem as the GPS Compact Flash card can be temporarily removed and replaced by the user because of the user initiated nature of RFID connectivity in the Proximity-Sensitive System design.

5.7.2 Technology Specific Issues

Implementing the Explorer also posed some Bluetooth-specific challenges. As stated earlier, the two PDAs used for prototyping come with built-in Bluetooth adapters and the Widcomm Bluetooth stack. Although both devices were already capable of supporting Bluetooth discovery through their own hardware drivers, source code access was not available to customise this code and direct the discovery results (context information) into the Bluetooth context file. To address this, it was decided that custom code would be written for the Bluetooth discovery function. The .NET Compact Framework (the platform chosen for prototyping), however, does not provide native support for Bluetooth development (Roof, 2002), and it only supports Bluetooth through the use of Microsoft's 'open-source' libraries. These libraries only work with the Microsoft Bluetooth stack and are incompatible with Widcomm stack. As a result, a third party library was required to develop a Bluetooth discovery function and Franson BlueTools (Franson, 2007) for Compact Framework version 2.0, which provides support for both Microsoft and Widcomm Bluetooth stacks, was chosen and purchased for developing the Bluetooth discovery function. This toolkit gave access to its source code (written in C#), allowing the incorporation of the code sections necessary for creating the Bluetooth discovery routine and passing the context information to the Explorer for updating the Bluetooth context file.

Like Bluetooth discovery, RFID development also faced many challenges because .NET does not provide native support for RFID application development. The Microsoft Developer Network (MSDN) online library (Caughey, 2004) recommends that RFID application developers use COM interrop or barcode class and serial class provided by OpenNET or hardware specific SDKs to resolve this problem. The decision was made to use hardware specific drivers and SDKs for two reasons. First, the SocketScan software which came with the Socket CF RFID reader and the Phidgets SDK which came with Phidgets reader already provided the functionality necessary for RFID discovery. Second, documentation relating to both the SocketScan and Phidget SDK suggests that the developers can create

RFID applications quickly and easily. This appeared to provide a better solution for implementing the proof-of-concept without being restricted by the limitations that are specific to .NET Compact Framework.

Once the decision was made to choose an RFID hardware specific solution, it became necessary to make a choice on the type of RFID reader that would be used for prototyping (Socket or Phidget). However, as the Socket and Phidget standards were incompatible, it was impossible to use a combination consisting of a socket RFID reader on one of the Clients (as sender's device) and Phidgets on the other (as recipient's device). Alongside this, due to the lack of a serial port on the rx3715, the Socket CF RFID could not be connected to the recipient's PDA rx3715. When an attempt was made to connect a Phidgets RFID reader to both Clients via USB ports, the CAB file installation produced a 'platform not supported' error message on both PDAs. The Phidgets Support and Discussion Forum (Patrick, 2008) explained that Phidgets CAB file for Windows CE only supports Windows Mobile 2005 or later versions, and it could not work on the Clients (PDAs) because they both have Windows Mobile 2003. This highlights a major standardization problem in RFID compared to Bluetooth and GPS: different vendors create different types of readers and tags, but they are incompatible and cannot work together in a single system. It is therefore important that developers choose RFID readers and tags that are compatible before starting the implementation of the Proximity-Sensitive System, and recognise the need to use the same type of RFID readers on all Clients. For the purpose of creating a proof-of-concept, it was sufficient to use Socket CF reader and SocketScan, and demonstrate that a Client is able to gather RFID tag ID from the current environment and use that information to support Proximity-Sensitive Messaging.

5.7.3 Deployment Issues

Deploying an application on a mobile device is cumbersome compared to deploying desktop applications as it requires a desktop computer, Microsoft ActiveSync, device .CAB file extractor (i.e. wceload.exe) and a mobile device. In addition, as these can vary in versions, processor power and type of code used in the application can cause serious compatibility issues when creating .CAB files and installing them on mobile devices. The mobile device cannot use an .msi file (used by desktop) directly, rather it needs a .CAB file for installing the application on a mobile device. The .CAB file is generated when the deployment package is created and this is then, in turn, installed on the target mobile device. Because mobile devices have different operating system versions in addition to their processor types, applications may need to create several .cab files in order to support these variations and versions. For example, the .CAB file created for the HP iPAQ h5550 also worked on the rx3715, but this does not mean that the same CAB can be installed on other PDAs (e.g. devices with Windows Mobile 5).

Although it was possible to deploy a .CAB file that was compatible with our HP h5550, the Bluetooth code failed to work on the device, producing an error message, "Failed to load Bluetooth driver". However, surprisingly, the same Bluetooth code worked on the rx3715 PDA. This is a device specific problem related to iPAQ 4150, 5500 series and 1450, and was resolved by copying Ipaq4150\BtCoreIf.dll and Ipaq4150\BtSdkCE30.dll to \Windows directory on the device (Franson, 2007). It should be noted that such problems may vary according to the mobile device that is being used and might not have the same solution across all mobile devices.

The SocketScan software for implementing the RFID discovery and context-update function failed to work with ActiveSync (version 3.7) that came with the PDAs. This is because it only works with ActiveSync 4.0 or later versions. While contemplating an upgrade and reading ActiveSync 4.0 related information, it emerged that ActiveSync 4.0 is not recommended for devices with Windows Mobile 2003 or Windows Mobile 2003 2nd Edition. There were too many problematic issues with ActiveSync 4.0 to go ahead with this version. The most problematic of these was that it does not have un-install function even when the installation crashes due to a lost connection. Thus, once installed there is no way of going back to its previous state, unless someone is prepared to find every file on the device and do a manual delete, not necessarily a safe option either. After devoting a great deal of time researching for information on ActiveSync 4.0 and later versions, it became clear that it would be sensible to avoid Active Sync 4.0 and upgrade to Active Sync 4.2, which only became available at later stage of the prototype development. Most commercial development projects cannot wait this long as their development time scales are short.

The GPS discovery routine was implemented using Franson GPS Tools (Franson, 2006) and the GPS implementation too faced many challenges at the deployment and testing stage. The deployment was unsuccessful from the Visual Studio 2003/ActiveSync environment due to a connection problem ("cannot establish a connection"). It emerged from the GPS Tools Support Forum that this problem only exists with Visual Studio 2003, and can be resolved if the application is deployed from Visual Studio 2005. At this stage a decision was made to upgrade the development tool from Visual Studio 2003 to 2005 (please note that Visual Studio 2005 was not available at the early stages of the implementation). By doing this, the development platform was upgraded from .NET Compact Framework 1.4 to 2.0 (Visual Studio 2005 comes with .NET Compact Framework 2.0). This left the proof-of-concept with an out-dated Bluetooth implementation. The Bluetooth solution that was created for .NET Compact Framework 1.4 refused to work in .NET Compact Framework 2.0. Much time was devoted to changing the Bluetooth implementation to use the correct libraries and code, and preserve the previous functionality in the new environment.

The next sections tabulate some development steps that could help to overcome of the problems in designing and implementing systems such as the one discussed in this thesis.

5.8 Recommended Steps for Developing Context-Aware Systems

It is acknowledged that designing context-aware systems for mobile platforms is challenging. Therefore developers who are implementing complex mobile solutions such as context-aware systems should adopt a methodical approach during their development process. Based on our experience so far, a set of guidelines are listed below to aid future development.

1. Identify a set of requirements for the system to be developed.
2. Decide on the context type(s) to be used for designing the system. Examine different types of context types, analyse their suitability in providing support for highly mobile and dynamic environments.
3. Make decisions on the context information necessary for developing the system i.e. will raw sensor data be sufficient or does this sensor data need to be interpreted and translated into more meaningful information before they can be utilised by Service Architecture.
4. In order to identify candidate technologies for gathering context related data, critique existing technologies, identify their pros and cons, and gather information on their usage among current mobile users.
5. Analysis characteristics of each available technology against system requirements and make decisions on suitable technologies for developing the system. Identify technologies for Client (user devices), context sensing and connectivity.
6. Identify a suitable development tool and platform for developing the system. This requires research into specific operating systems and hardware specific software used on Client devices (various mobile devices). For example, applications written for Windows Mobile 6 may not necessarily work on Windows Mobile 6.1.
7. For each technology chosen, search for information on system specification including the requirements, limitations and issues faced by

current developers. It is imperative that compatibility issues and limitations of hardware and software to be used are thoroughly investigated prior to purchase as manufacturers are unable to cover all the details for each mobile device.

5.9 Summary and Conclusion

The essence of the discussion and analysis in this chapter is that proximity sensitive messaging for mobile devices, as a distinct and viable option, *can* be achieved through the implementation of a Proximity-Sensitive System Architecture that builds on the capabilities of common consumer devices. This was demonstrated through the creation of the proof-of-concept prototype that encapsulated the novel ideas put forward by the thesis, on a range of advanced (at the time of development) and commonly available devices, platforms and tools within the many possible variations, to implement the architecture. This highlights that Proximity-Sensitive System Architecture is not a theoretical model but can be implemented using off-the-shelf hardware and software.

The implementation was not without challenges; convergence, technological and deployment issues surfaced at various stages, exacerbated by issues of resource constrained devices, technological convergence and lack of standardisation found in some of the technologies used. The implementation process helped to understand four important aspects of prototype development for proximity based systems:

1. Mobile application development is very different to desktop application: it requires in-depth knowledge of the devices, development platforms and tools.
2. Lack of standardisation between different vendor specific devices, platforms and tools limits application support to specific devices.

3. The unpredictability in development resources and their interdependencies can make the implementation process extremely hard to plan for and schedule.
4. The Proximity-Sensitive System Architecture is not a theoretical model, but one that can be successfully implemented using existing but carefully selected devices, platforms and tools.

It is hoped that these lessons learnt by the implementation process prove useful to future developers interested in developing systems for mobile platforms. In the next chapter, the proof-of-concept is used for reviewing the design concepts of Proximity-Sensitive Systems Architecture.

Chapter 6: Evaluating Proximity Based Approach Users View and Vision, and Analysis

6.1 Introduction and Overview

This chapter evaluates the proximity based approach developed in this thesis for supporting context-awareness. The evaluation process discussed here is entirely qualitative and was carried out in two ways: user study and reflective theoretical analysis. The user study is intended to contribute to a better understanding of the proximity based approach, both in respect to personal communication, and also to the broader notion of proximity-based services. This it does through ascertaining how users understand and interpret proximity and their views on electronic systems that support proximity-sensitive communication. The theoretical analysis provides an overall assessment of the Proximity-Sensitive System Architecture from a designer's point of view, and offers insights into the opportunities and limitations of proximity based systems. Together, both evaluations offer insights into the design of the proximity-based system.

6.2 User Study

The data for the user study was collected through semi-instructed interviews with individuals complemented with a technology probe approach (Hutchinson *et al.*, 2003). The sample for this study was a purposive sample (Trochim, 2006), with participants chosen from varied backgrounds broadly representing a cross-section of mobile device users, and from whom it would be possible to obtain information pertinent to the research.

Background details of the sample population is summarised in Table 6.1. The sample consisted of ten participants (five male and five female) aged between 14 to 65. In the table they are grouped under the age brackets 14-20, 21-30, 31-40, 41-50 and 51-65. Some of the participants were from professional backgrounds, some from trade and administrative backgrounds and some were school pupils. One characteristic that the sample had in common was that they all owned and used at least one mobile device and had access to SMS messaging, which was considered to be the closest commercially available service to the proposed design. They are thus familiar with several, but not necessarily all of the features demonstrated in the user study. Participants were anonymised, and are referred to in the thesis by their initials. All criteria in Table 6.1 are self-reported, so the data is dependent on their interpretations. For example, what constitutes of ‘low’, ’medium’ or ‘high’ mobile device usage to one person may be only relative to their experience of others.

Interviews with each participant lasted between one to two hours and focused on eliciting information on the participants’ understanding of spatial proximity, and how proximity was used in their current day-to-day communications, both personal and professional, as well as their expectations for the future. In addition, participants were given the opportunity to express their thoughts on proximity in comparison to location in order to help them identify similarities and differences.

The questions used for interviewing the participants are listed in Appendix D. However, it is important to note that these questions were only used as a guideline to initiate discussion and encourage participants to share their thoughts.

Participant	Gender	Age	Background	Mobile-Device Usage
CT	F	14-20	Student	High
ST	F	14-20	Student	Low
AK	M	14-20	Motor Trader	High
AA1	M	21-30	Office Administrator	High
AS	M	21-30	Auditor	High
CG	F	31-40	Homemaker	High
CS	M	31-40	Radio Design Engineer	High
SSJ	F	41-50	Service Delivery Manager	High
AA2	M	51-65	Hospital Consultant	Medium
HH	F	51-65	Retired Librarian	Low

Table 6.1: Participant Background Data

Following this semi-structured interview, the technology probe based evaluation was carried out using the proof-of-concept prototype created in Chapter 5. Technology probes can be described as simple, flexible and adaptable technology with three main goals:

'The social science goal of understanding the needs and desires of users in a real world setting, the engineering goal of field-testing the technology, and the design goal of inspiring users and designers to think about new technologies.' (Hutchinson *et al*., 2003:18)

The technology probe's purpose goes beyond design, in that it lets users consider systems with new features and allows them to explore the world (their environment) from a different angle well before a design is developed into a system or prototype that is ready for field trial. It also allows researchers to collect data about their users' vision (i.e. where this system can be useful) on the Proximity-Sensitive System, their interest and concerns, and, more importantly, it helps to understand what users want from such technologies rather than simply providing them with what researchers or designers think is best for them.

The proof-of-concept was explained and demonstrated to participants in some detail before they were asked specific questions about proximity-sensitive communication. During the demonstration, participants were given a walkthrough (see Appendix E) for each of the two example scenarios discussed in Chapter 1. Following this, the interviewees were encouraged to express their understanding of the system and envision such a system in their own lives. They were asked questions about their interest in the system, their needs and any concerns they may have in using such a system in their daily lives. It should be noted again, however, that, and in line with semi-structured interviewing practices, these questions were only used as a guideline to initiate discussion and encourage participants to share their thoughts.

6.3 Understanding Proximity: Users' Views

The information collected from the interview was read and quotations that illustrated salient points about the research's focus were chosen. Following this, an affinity diagramming process (Beyer and Holtzblatt, 1998) was used to categorise the chosen quotations. Affinity diagrams allow categorisation of large amount of data into logical groups based on the perceived relationship between ideas. The quotations were first placed into colour-coded hierarchies before arriving at three high-level logical groups: what proximity and location mean to users, proximity in present day communication and current communication methods – users' views. These three groups are discussed in detail in sections 6.3.1, 6.3.2 and 6.4 respectively.

6.3.1 What Proximity and Location Mean to Users

The interview data suggest that the participants had a good understanding of the term proximity. They were able to explain what it meant to them, and some were even able to provide examples of interpersonal or public communication scenarios from their own lives that related to spatial proximity. The majority of them saw proximity as ‘closeness’ to something or an ‘area around’ something. Some of them indirectly included scalability in their definition by stating that proximity could tell them how close or far they are to something:

“…. tells me how close I am to something; a point, place or a thing. It makes me understand where I am in relation to another thing” **[SSJ]**

Yet another participant defined proximity as an area around a place or a thing:

“Proximity is an area around a point or an object.” **[AS]**

Some participants clearly showed some understanding of proximity beyond places or a point in space, expressing it as a relationship between two things. AA2, who works as a Hospital Consultant, was very clear in his mind that proximity cannot be used independently without relating to another thing. He discussed proximity in terms of a relationship between himself and other entities in his life (home, hospital ward, his patients, desk, etc.):

“Proximity is a relation between me and something; me and the car park, me and my home, me and my patient, me and my ward, me and my desk …… ” **[AA2]**

Although AA2 did not explicitly state that proximity can support relationships between two mobile elements, he included two mobile entities in his examples (i.e. the relation between him and his patient). This shows that he understood that a proximal relationship can exist or can be established between mobile entities.

Only one participant out of the ten interviewed explained proximity as closeness between two entities that might be static *or* mobile. In his words, proximity:

*"… is closeness in reference to a fixed point or an object." [**pause]** "I guess nothing is stopping us from providing closeness in relation to two stationary or mobile objects. Although, not sure whether it is generally used in this context."* **[CS]**

What is interesting is that after defining proximity in terms of static entities, he paused and reassessed, then explained that there is more to proximity than fixedness: *"two people could be walking and still maintain their closeness."* From this, it was clear that he was considering the situation as two moving people in proximity to each other.

The term 'location' was then considered, and participants were asked to collect their thoughts on location and compare these with their thoughts on proximity. Most participants thought location was a point in space, or place. One person said it was where something resides. Some thought of it as a place (e.g. a hospital, shop or home) where something is provided. However, they all said it was different to proximity. CS, who is a Radio Design Engineer, even stated that *"proximity is relational where location is absolute. Location tells us whether we are there or not, and proximity tells us how far we are from something."*

He also raised an interesting question on relational location and proximity, and he wanted to know the difference between them. Then he answered his own question: *"… we cannot define relational location in terms of two people who are walking can we?"* **[CS]**

Although it is not the intention to provide definitions for the concept location or relational location, the data illustrates that users clearly understand that proximity offers more than just relational location. As stated in earlier chapters, and as acknowledged by one of the participants (CS) as a concept offering a distinctive set of characteristics compared to location, and allows to move away from ‘fixedness’ if necessary.

6.3.2 Proximity in Present Day Communications

The majority of instances of current activity in the data collected for this category came from the use of non-digital media; i.e. people mainly described situations where they used paper and pen or other people to deliver messages. Most participants said they talked to people when they saw them, and left notes, post-its or messages with other people when the message recipient was not present. Eight out of the ten participants also indicated that they frequently used non-proximity driven techniques, such as e-mail and SMS, for delivering written messages, including in situations where the message recipient was thought to be busy.

Almost all of the participants talked about leaving messages in the kitchen (e.g. on the kettle or fridge). A few said that they leave messages on their land-line telephone at work or at home. They all thought their messages were relevant to those contexts, and that these messages allowed more effective and situated interpretation and use (Rachovides and Perry, 2006). Moreover, they thought their message recipients were likely to come across these entities and thus there was a high likelihood of message delivery. One person said that he often finds notes on the fridge if his partner is out for the evening: *“Food in the fridge, microwave full power for 2 minutes.”* **[AA2]**

He added that his partner knows that the fridge is the most appropriate place for leaving a message about dinner. If the message was left somewhere else in the house he might not even see the message before he starts preparing his dinner. In the same way, some participants reported that they stick hand written phone messages to land-line telephones. For example, ST said that she leaves messages for her mother saying “*Nick called and will call back at 8 pm*.” It is interesting to note that people are using land-line telephones when they are at home, and are seeing them as informationally-interesting entities for use as a site of physical interaction.

In a variant on this use of paper as a medium, the younger participants described situations where they used their siblings to pass messages to their parents. From the conversation it was clear that these types of communication were quite normal in their lives.

"If I am meeting my boyfriend after school, or might be late that evening, I normally ask my sister (who is also in the same school) to let my mum know. I always give details on where I am, whom I am with and what time I will be back." **[CT]**

She went on to explain that her message can then be passed on when her sister meets her mother. She added that she tries to avoid phoning her mother unless it was urgent.

"I don't phone her because she works during the day, might be with a patient or driving." **[CT]**

AA2 discussed many situations and techniques used in the hospital. He said that he and his colleagues used written notes, white boards or other people to deliver such messages. In an emergency, doctors and nurses are alerted via the bleep system and were usually expected to report to a particular department (e.g. the Accident and Emergencies reception area). In the department they used a whiteboard to display a task list with staff details (matching tasks against staff names), which gets updated when the doctors and nurses report to the department. In some cases, a person stands with a paper based task list to provide the same type of information to the staff when they arrive at the scene. The information is however, irrelevant if they fail to turn up where and when they are needed, and so this information, and the work going into its creation, is wasted.

AA2 also cited several situations where patients' notes (written or electronic) are transferred by people between departments (e.g. Orthopaedics and X-ray), and between doctors when they perform a handover of patients at the end of their shift. For example, doctors and nurses often go through their patients' notes to gather specific details, and events that happened during the previous shift. In this, information is related to the patient, and is required when the doctor or nurse is in close proximity to that patient.

In contrast to the hospital environment, CS, the Radio Design Engineer who has a good working knowledge of technology, had a different view about proximity-based communication. He said that his work environment has changed in the last five years, and has witnessed the evolution in the use of mobile and wireless technologies: *"There used to be a time when we all leave messages on desks and computer screens but I can't remember the last time I used a post-it note at work"* **[CS]**

He then went on to explain the reason why they no longer leave messages on desks or computers at work:

"With mobile phones and wireless connections, people are free to move. They have their phones with them all the time and they can be contacted almost anytime. Moreover, people tend to work from home, go away for meetings or work away in another office. So if they are not at their desk, I normally assume that they are away for the day, working away from their office (it is safe that way), I don't leave messages on their desk. There is an element of uncertainty, not knowing when this person will return. It was very different when we used our desktop computers. People have to return to their desk to work, and it was worth associating messages to their screens or desks. This is not the case anymore. They can do their work from wherever they want. For this reason, I normally send an SMS, email or even try to phone them. We all have work phones, so people are happy to respond." **[CS]**

These comments by CS support what has been described by other researchers: “Gone are the days when work took place at the office and family life at home. For some, work implies several work sites …..” (Schmandt *et al.*, 2000). Yet it also simultaneously shows the difficulties in providing messages in the right place and the right time with current mobile technology.

The data from the interview gives more information about the way people communicate in their workplace, and shows that they are no longer tied to their desks. The implication is that research can no longer focus solely on proximity relating to fixed places or static entities (e.g. office desk or computer) to deliver messages and, further, that context-aware communication has to move beyond ‘fixedness’ (i.e. locations and places).

It should also be noted that when asked to provide likely scenarios for proximity-driven communication in their personal lives and professional practice, not all participants saw such communication as proximity driven, or saw this as an important or noticeable feature of such communication. Some of them had to be prompted with examples from the interviewer’s own experience. This shows that although users understand proximity and have used proximity driven communication in their lives, they have not given any thought to the underlying concepts of such forms of communication.

6.4 Current Communication Methods: Users’ Views

The interview data reveals that participants rely on non-digital methods for delivering proximity-sensitive messages. As CS pointed out, current technologies have not evolved enough to support such services. This section examines how participants feel about using existing methods of communications, whether they are satisfied with current solutions, what their concerns are, and what they indicated was their vision for technology-based solutions.

6.4.1 Social Issues Relating to Current Communication Methods

This sub-section focuses on current communication methods for proximity-sensitive services. The aim here was to understand how people were communicating, more specifically, the type of media they used. There was a great deal of variation in the way people responded to this question. As few participants were content with what they have; some picked and chose communications methods to suit their needs, with others frustrated with the current technologies that prevented them from doing what they really wanted to do (e.g. to have the freedom to leave and access information in relation to entities of their choice). Nonetheless, they all acknowledge that they have been in situations where they wanted to send or receive messages in relation to a place, object or a person.

HH (a Retired Librarian) thought that we as a community already have what we need to communicate socially in personal and professional lives, but what we lack is the understanding of how and when to use them. She went on to say that mobile telephones were introduced so that we can move around and still communicate when we need to. She explained that the word ‘need’ is currently being overexploited here, and said that we overuse these technologies without thinking of what the other person might be doing or where that person might be. She believes that this should be seen as a social problem rather than technology issue or limitation:

“We don’t need a new technology to help us with this, what we need is awareness of social expectations and manners.” **[HH]**

This quote demonstrate that there are people who feel that technology can be socially and contextually intrusive and the technology users should be in HH’s term *“educated to use them appropriately rather than keep introducing new technologies.”*

6.4.2 Proximity-Sensitive Services in Outdoor Environments

The current techniques for delivering proximity sensitive messages are paper based and are generally intended for indoors, with concerns being raised and inadequacies highlighted particularly for use in outdoor environments. CG commented, *"posters are used in public places to convey information but it is not personal."* Graffiti too is used in outdoor environment, but it is not a practical solution for personal messaging or dynamic information delivery because the information content in the latter changes very often, as well as the legal and cultural concerns that it raises as a medium.

Let us now consider a personal messaging situation discussed by CS, the Radio Design Engineer. He said that his wife often leaves sticky notes in the kitchen with details on where they are (e.g. *"I am taking the children to the park, will be there until 6pm"*), and went on to explain why leaving a post-it message in the kitchen is not appropriate despite her use of them. For instance, there are days when he comes home and finds a message saying that his wife and children are in the park or at a friend's house. Sometimes she asks him to meet them there, but he might have already passed the park or this friend's house on his way home without knowing that he has to come back to that place again:

"By the time I read the message in the kitchen I am already home, I have to get in the car and drive back. Waste of time and effort. It would be nice to have this message when I am close to the park or our friend's house."

He then elaborated on this, noting his concern in using SMS or a voice call (on a mobile phone) in such situations. He said that if he was already in the car when his phone delivers this SMS (alerted by a beep), it is very unlikely that he will have a chance to stop and read that message. In addition, he also said that he does not answer the phone when he is driving, making this solution problematic:

"I don't know what the message is about so I might think the message can wait - you can't routinely stop and read messages. Therefore, I will only get the message when I get home. Here, there is no difference between the post-it in the kitchen and SMS or voice message." **[CS]**

He then began to think about a solution to this problem. In view of his professional background (as a Radio Design Engineer), it is perhaps not surprising that he wanted to find a technological solution. He pointed out that currently there is no system to provide such personal services and discussed additional parameters (e.g. time) that might need to be looked at to make the system deliver messages where they are expected to be most relevant:

"However, if there was a way to deliver messages when I am in my home town, a few miles away from home, then I know that the message is relevant to my current area. In addition, I can be fairly certain that the message is from my family. So I will make an effort to check before I drive home. Of course we need a time stamp. If I am driving home after 6:00 pm there is no point in me getting the message. My family would have left the park at 6:00 pm." **[CS]**

This highlights the fact that some of the messages need more than a proximity relationship to deliver information where they are expected to be most relevant. In the above quotation, time also plays an important role as the message is not relevant to him after 6 pm, that is, after CS' family leave the park.

6.4.3 Insecure, Ephemeral and Unaccountable

As the data illustrates, paper-based methods for proximal messaging such as post-its notes are usually physically stuck onto or placed on top of things. They are not private or secure (i.e. not firmly fixed) and, therefore, can easily be misplaced or lost. As a consequence, messages written on them may not be delivered appropriately (in a relevant context) or worse, not delivered at all. Furthermore, the participants were concerned that there was no proof to verify that they have actually left this message. In contrast, they pointed out that they could go into the 'sent' box in SMS or e-mail and provide proof for sending their messages if necessary. In paper-based messages there is no such facility, hence the message sender has no way of proving that he or she has actually sent this message to make their action accountable to others. Nevertheless, the ephemeral nature of paper-based messages can be considered beneficial as they are not going to be present for ever to clutter the environment with content. In contrast to paper-based messages, electronic systems do not usually lose messages easily, but they do face issues relating to clearing or archiving the messages once they are delivered to their intended recipients.

It was also identified from the interview data that participants are quite happy to use paper-based techniques at home. Some of the participants stated explicitly that they use post-its at home and found them a reliable way to communicate with others in the household:

"…In my house, post-it notes are generally safe and people get their messages without any problem." **[ST]**

It was clear from the interviews that some participants did not like the idea of using mobile devices in situations relating to their home, and more specifically, to send or receive messages when it is relevant to their home environment. This indicates their interest to keep mobile devices away from home, so as to avoid intrusion into their personal family life. This could be interpreted to show their interest in keeping their private life separate from their social and business schedules.

In summary, this set of interviews investigated a number of factors associated with use and adoption of current communication methods. The answers varied greatly, while some participants were content with current methods, the others showed interest in using more intelligent services that would enable them to communicate where and when it is appropriate. In addition, the answers highlighted that current support for context related personal messaging is rather limited in outdoor environments.

6.5 Proximity-Sensitive Messaging: Users' Vision

In order to support the argument put forward by the thesis, the Proximity-Sensitive System was demonstrated and explained to potential users, showed them what it is and what it is capable of doing. This allows potential users to envision this system in their daily life and discuss the benefits and concerns. In the user study, this is achieved through the use of proof-of-concept prototype; the implemented prototype is used as a technology probe to communicate with potential users. It gave the opportunity to demonstrate system features even when all its elements were not enhanced for performance. In addition, it enabled the participants to envision the system and discuss freely about the technology and its use (Schmidt, 2002). The information collected during this user study is categorised and analysed in the following sub sections.

6.5.1 The Potential Role of Proximity-Sensitive Services

It was clear from the interview data that all the participants understood the proof-of-concept prototype and that they were able to come up with their own scenarios where it would be useful. For example, a teenage student said that she could use it for sending information to her mother when she is at school, showing that she understood the system well:

"My sister is studying in the same school. Some days my sister goes home early and I have after school activities. If I want to send a message to my mum, say… please pick me up at 5:30 pm near the front gate, I can tag it to my sister. My mum will get the message when she sees my sister. I don't have to go and find my sister in the playground to pass this message. More than that, I don't have to

bother my sister to pass this message. I don't even have to worry about it, i.e. whether she will remember to pass the message. It will be done automatically and delivered when mum is not at work." **[CT]**

In this instance, CT was referring to a situation where all entities involved were mobile. Furthermore, she was aware that her sister was not actively involved in the communication, but that her device would be used as a vehicle to carry the message to her mother.

It was interesting to learn that even the participants from non-technical backgrounds understood the proximity-based approach to communication very well. AA2, who does not consider himself a technology expert, was able to explain a situation from his life. He said that it would be useful for him to have such a system to assist with his ward rounds. It was clear from the way he went on to discuss his interest in tagging information to patients rather than hospital beds and wards, that he understood the prototype. More precisely, he was able to recognise that the prototype supported tagging in relation to mobile (e.g. patient) and static entities (e.g. a bed, a ward). He further discussed the use of such a system in other departments (X-ray, scans etc.) and wards in the hospital.

"…I would like this information to be available when we are with the patient not necessarily in that particular ward. We deal with patients, they move around from ward to ward, in some cases bed to bed. The information must move with the patient. Some times they go for scans and test, and would be useful if the relevant information is passed onto the medical staff dealing with these patients." **[AA2]**

Although providing medical solutions is not the intention of this thesis, this statement illustrates that people are interested in relating information to people. In addition, it helps to point out a real situation where all entities involved were mobile: medical staff and patients.

Some of the participants showed interest in delivering messages in places such as shopping centres, hospitals, shops, etc. Although, comMotion has already addressed this to a certain level, the participants have appreciated the flexibility to choose a specific area for message delivery. For example, participants CT and AA1 have noticed its flexibility, and discussed how it might impact on their activities:

"If I have arranged to meet someone in the shopping centre (WHSmith), I get there but the other person is running late. I also have to pop into Woolworths. So I can leave a message at WHSmith 'I am off to Wooly, will be back in 5 mins'. Even better if I can cover the Woolworths and leave the same message. If that person is passing Woolworths then he or she may decide to meet me there, saves us coming back to WHSmith." **[AA1]**

"I might go to Gap and find something nice. I can leave a message for my friends or sister. Something like check this out! What is better is that I can even cover the whole shopping centre if I want. So they don't have to be in Gap to receive this message." **[CT]**

From the above two quotations, one could say that people saw proximity as a concept that not only helps to find themselves in relation to mobile entities but also in relation to places and geographical coordinates. What is interesting about this is that the Proximity-Sensitive Service is supported using not only Bluetooth sensors but also absolute location GPS sensor.

The participants also commented on situations that take place outside personal communications. Some of them pointed out that the prototype could accommodate services relating to restraining orders, shop-lifters, medical information on serious conditions and referral letters:

"If convicted people are not allowed to go near someone or place, they could be tagged. I guess this would not be considered as a privacy issue, he has already committed a crime and is serving a sentence so we need to alert people." **[AA1]**

"Alert shops about shop lifters….." **[AA1]**

"Tag medical notes to people who have serious medical conditions." **[AS]**

"Referral letters going from GPs to hospital doctors about patients. Instead of patients carrying this information we could provide an automatic delivery, less chance for losing such information." **[ST]**

These examples described by the participants show that they had an understanding of the system and how it could be used in their lives and incorporated into a wider social and cultural context. They were able to explain situations where communication was driven by proximal distance between mobile entities, mobile entities and geographical coordinates, or between mobile entities and places. Some of the participants were even able to describe Proximity-Sensitive System's use outside the application being demonstrated to them (context-aware personal messaging). In addition, they also commented on its flexibility, how it allows users to choose entities to leave messages, and proximal distances for delivering their messages. For example, one thought it might be worth telling her friend to visit Gap when they are in the shopping centre rather than waiting for them to visit Gap on their own initiative. They understood that such form of scaling could be supported by a Proximity-Sensitive System.

6.5.2 The 'Right' Device: Screen Size, Readability and Cost

During the interviews some concerns arose about the mobile devices that were going to be used to implement such systems on. One commonly referenced concern was due to the fact that mobile devices come with very small screens, and interviewees thought that these would not be suitable for their purposes. Another issue was that some participants were concerned that mobile devices are still expensive, and every time a new service is introduced they would have to buy a very expensive new mobile device. These concerns are discussed in more detail below.

As noted in the last section, AA2 identified how medical staff could use such a system for ward rounds. He also developed this further, stating that they usually go as a group when they do these rounds, a consultant, junior doctor, nurse and possibly medical students, and that having such information on a single mobile device screen or on individual mobile devices could be problematic:

"I would like to stress that I don't like this information on a small mobile device such as a mobile phone. Their screens are too small for our purpose, and we will not be able to see the information or pictures in detail. Additionally, we will all end up carrying our own devices, viewing the information on our individual screens. During the discussion we might be on different pages of the notes, causing confusion, and in some cases misunderstanding and errors. We want to avoid this." **[AA2]**

Although, supporting such services is not the primary focus of this thesis, this information is included to highlight the fact that the Proximity-Sensitive System Architecture has the potential to provide services outside interpersonal messaging as required by the hospital consultant. Furthermore, tailoring services to deliver information on a larger screen, this is not going to cause any major problems to designers.

A number of participants commented that mobile devices are still very expensive and, at the same time, continually changing. Further, new services are always introduced on latest devices without any backward compatibility. Therefore, if they wanted to have access to the new services they would have to keep buying new devices. They were concerned that to use proximity-sensitive services they might need to buy a very expensive state-of-the-art mobile device to get access to proximity-sensitive messaging service:

"My concern here is whether we need to buy yet another device to get access to this service. Mobile devices are not cheap. As it stands, if I want a new feature then I have to buy a new device. I don't want to keep buying new devices and spending my money. I have already done this several times with GPS and MP3 phones." **[CT]**

Another participant wanted to know whether he will have to buy additional receivers to get access to proximity-sensitive services. He commented that his main concern is that he may end up buying *yet another* device to accommodate this new receiver:

"I bought GPS receiver and ended up changing my phone to Nokia 6600 to have Bluetooth. Now they have introduced Nokia N95 phones with built-in GPS receiver. If I don't want to carry too many devices then that is what I need. But will it stop there?" **[AA1]**

Although cost does not play a major part in design requirements of this research, the data suggests that the decision to make use of existing Environmental Sensors was the right one. This was done for two reasons. The first was the cost issue raised by most participants: developing a rich and pervasive sensor infrastructure and custom designed devices is far more expensive than simply using an existing set of sensor resources and devices. The second reason was that there are increasing numbers of sensors that can detect signals available on mobile devices. These are already quite pervasive within the environment, which means that people do not need to add to their existing complement of devices that they carry around with them. As noted earlier, proximity-sensitive applications can be developed relatively easily on top of the currently commercially available devices without costly and complex customisations, and the reliability problems that these are likely to introduce.

6.5.3 Accuracy as a Concern in Proximity-Sensitive Services

This section highlights the importance of accuracy in Proximity-Sensitive Systems, and the concerns raised by the participants about this. Some of the participants wanted to know the exact area in which the message would be delivered. One participant said that she often meets her friend at the local train station when they go to London for shopping. She added that when she gets to the station, if her friend is not already there she could sort out the tickets without waiting for her friend to arrive. In such a situation she said that it would be better for her to leave a message for her friend, and be certain that it would be delivered as soon as she walks into the station. However, she thought that the Proximity-Sensitive System might not provide the accuracy necessary for this, to deliver the message when her friend gets to the station. If the message is delivered late then her friend may decide to do the same thing (i.e. buy another set of tickets):

"I don't want the message to be delivered when my friend is driving, at the same time I don't want my friend to miss the message simply because he is near the entrance and not near the ticket office." **[SSJ]**

The participant also noted that SMS is an instant delivery system and, therefore, the message delivery does not depend on her friend's action: "*In Proximity-Sensitive System, the message delivery relies on my friend's action and thus takes the control away from me.*" Of course, messages may never be accessed by her intended recipient if she does not come into the message delivery area. In the proposed design, this is intentional, and should not necessarily be seen as a disadvantage of the system. Messages are intended to be connected to people, places and things, and if these physical entities are not encountered, then their content is unlikely to be highly relevant.

In the other situation cited above, CT asked whether her sister has to be very close to her mother to deliver the message that she tagged onto her sister's device. She was concerned that if her sister was in her bedroom and her mother was in the kitchen, the message may not be delivered because in this case, her sister and mother will be unaware of the message. This shows that CT understands that the message sender has to make appropriate selection of sensors, and select a suitable distance for message delivery. This is the very reason three different types of sensors with varied levels of coverage were incorporated. Further, this reinforces that these different levels of proximity should be clearly made known to users so that they are able to make appropriate associations as to their use or meaning.

6.5.4 Privacy as a Concern in Proximity-Sensitive Services

In the interview, several participants raised issues relating to privacy. CG, a full time mother and an active member of her son's school welfare committee, pointed out that although paper-based techniques are quick and easy methods for leaving messages, they are not without limitations. In her opinion, post-its are very useful when she wants to leave messages at home for her husband. However, she had concerns in using such messages in public places, such as when she is arranging a function at school:

"If I am organising a summer festival at school and I want something set up somewhere say on Table 23, in Class A or tennis court. I might need to leave a message for someone who is setting up that activity. However, I would be very reluctant to leave a note or post-it on the table or classroom door because it can easily get lost or misplaced, as a result the stall will not be set up or even set up in the wrong class/on the wrong table; there is no proof to say that I have briefed the

person responsible for this job, and everyone will have access to my message – no privacy." **[CG]**

Although Proximity-Sensitive System was able to offer privacy CG said that it does not guarantee delivery. She went on to discuss her interest in a service that could provide this support:

"It would be nice to have something that guarantees delivery such as SMS so I can consider it done. At the same time it delivers to the right person rather than anyone passing Table 23, Class A or tennis court." **[CG]**

Interestingly, although SMS does not guarantee delivery, some interviewees believed SMS was reliable: the majority were not aware that the SMS server can fail to forward messages to their intended recipients. Even when made aware of this SMS limitation, they commented that SMS messages were sent electronically and, therefore, they were more reliable than paper-based techniques. Some even claimed that they had never experienced any problems with SMS.

The participants also raised concerns in using proximity-based tagging. Some of them pointed out that proximity-based system allows to attach messages to people in a way that might make them feel uncomfortable. The person to whom the message is attached is likely to be completely unaware that they have been tagged or labelled, and may have no mechanism to remove such messages (other than to turn off or put down their devices) even were they to realise that such a message had been left. The user study participants pointed out some situations where people might exploit such services.

"There is a potential to carry incriminating or negative messages/information. For example, boy in a pub, being explicit with a girl, a friend takes a photo tags it to the boy's device and sends to his girlfriend. The boy will have some explaining to do when he meets his girlfriend - message will be delivered when he meets his girlfriend." **[AA2]**

The majority of the participants said that they could think of many instances such as this. Yet this is not novel, and is a problem faced by many messaging systems (e.g. ‘kick me’ message). However, with the Proximity-Sensitive System it may seem rather odd that the person himself is being used as a carrier to transport incriminating messages about him or herself. In the above case (AA1), the boy is being used to carry his own photograph of evidence. One of the participants interviewed wanted to know whether he could selectively deny permission or allow permission only for a certain group of people to tag messages to his device. This is a somewhat difficult problem to resolve. One solution may be to allow people to examine and edit messages that are associated with sensors that they are responsible for, although this can itself be open to abuse. Although it is not referenced in user interviews, it is also likely that were the system to become widespread, an etiquette could develop around such proximity sensitive messaging and other services that provide a set of acceptable forms of behaviour.

6.5.5 Service Reliability, Cost and Expectations

In this section, issues relating to message delivery are discussed. The participants were keen to find out whether message delivery could be guaranteed by the envisaged system. Several participants compared proximity-sensitive messaging with SMS and e-mail. It was obvious from their conversation that they were quite comfortable with those two messaging systems. In addition, they thought these systems guaranteed delivery. However, in practice this is not necessarily true. Some of them even thought that the messages in the sent folder (for SMS and e-mail) operated like a receipt for their delivery (i.e. a form of confirmation for delivery). Although what they think or believe about SMS and e-mail is not directly related to this thesis, the findings do have an impact on the views and expectations people may have about the Proximity-Sensitive System. If they believe that they already have receipt functionality in current messaging systems, it may be only natural to expect this in future services.

Some participants acknowledged that messages will only be delivered if the recipient is in close proximity to the tagged person, place or an object. Unlike an SMS or an e-mail that is likely to be delivered irrespective of its recipient's location, the proximity-sensitive messages may never be accessed by their recipient if users with messages awaiting them may not come into range of their tagged entities. They were concerned about this uncertainty, not knowing whether the message was actually delivered or not. One participant said that she would like to know if her message was *not* delivered:

"If I send an e-mail or SMS I can consider it done however, with this system I don't get any visibility. There is no way of knowing whether my message was delivered or not. I would at least want to know if my message was not delivered." **[SSJ]**

Although there is no delivery guarantee with SMS or e-mail, the participants thought these messaging services are generally reliable. During the interview it was brought to their attention that the proximity-based delivery is intentional. This led them to think about cost implications. They wanted to know whether mobile operators will be charging for undelivered messages. Although the service is far from its commercial launch, it is useful to highlight the concerns people have in using such systems. Several participants pointed out that SMS is charged for sent messages only. What most of them did not realise is that they are charged when their SMS is accepted for delivery by SMS gateway and not when it is delivered to recipients (BT, 2005). Further, such acceptance does not guarantee message delivery. However, in Proximity-Sensitive System message delivery is triggered only when the recipient (his or her device) is in close proximity to the tagged entity, and thus adds a measure of uncertainty regarding delivery. If for some reason delivery is not triggered, then the message will not be delivered. The participants were reluctant to pay for those undelivered messages, a consideration for operators to bear in mind were they to launch such a service.

In summary, the participants were able to understand the proof-of-concept prototype and its use in their daily lives. The majority of the participants were able to highlight the potential of Proximity-Sensitive System outside personal messaging services. They also showed interest in the way the system allowed them to tag messages on to objects and places of their choice. Some of them even discussed its scalable feature, flexibility to define areas for message delivery. However, the participants had concerns regarding accuracy, reliability, privacy and cost. Some of these concerns warrant further work if this system were to be commercialised.

6.6 Reflective Analysis on the Proximity-Sensitive System Architecture

The investigation undertaken in this thesis is not about replacing or optimising existing systems, but rather about introducing novel ideas, namely: proximity and technological convergence, to the design and development of context-aware systems to offer more targeted, enhanced service options to users. Thus, the experience and lessons learnt during the course of development of the Proximity-Sensitive System could serve as a basis for future research and investigation into proximity-sensitive services. For this reason, a critical analysis is provided below to cover features and experiences that have not been addressed in the user study.

- The proximity-sensitive architecture allows systems to expand and shrink dynamically with minimal administration, i.e. it allows entities to be added and removed as needed for coverage and communication purposes. In this, it is different to Active Badge (Want *et al*., 1992), Cricket and ParcTab (Want *et al*., 1997) which rely on purpose built infrastructures whose components have to be deployed and maintained. The approach adopted in this thesis utilises commodity mobile devices with sensors and existing Environmental Sensors to take part in the communication by simply being around in the environment. In addition, it utilises passive RFID tags that can be stuck to any object to be included in the system, providing for easy installation and maintenance.

- The proximity-based Architecture presented has the potential to extend beyond proximity-sensitive personal messaging. This was recognised by many of the participants interviewed in the user study, who pointed out its use in offering medical and criminal information services to target audience. Further, systems that provide support beyond a single application are considered to be the way of the future for mobile ubiquitous computing (Schulzrinne *et al.*, 2007) and the proposed architecture supports this vision, which handles complexity by separating the various components of the Service Architecture. It keeps proximity sensing separate from routing and interface and provides context gathering through a separate and purpose built software component that is installed on the mobile device. This Proximity-Sensing component is designed to work independently without interacting with the application, thus gives the freedom to develop various applications on top of the same Proximity-Sensing component.

- The user interface to Proximity-Sensitive Systems should allow users to make better informed judgements about the constraints of the various sensors (e.g. robustness and range of signal reception) that they are utilising. This is not something that is naturally visible to users, particularly as these signals are wirelessly transmitted and sensors are often invisible. These constraints need to be made apparent in the interface of the mobile device in some way so that it is possible to determine these constrains, and so to allow users to use and interpret the system appropriately. Although this could be done explicitly, through user manuals, or on-screen instructions, this could also be achieved by allowing the system's users to easily visualise the constraints of the sensors, for example, showing them when GPS signals are lost when inside buildings.

- Placing or locating messages is not a simple matter: senders may not fully realise or understand where or on what they are placing a message. Similarly, recipients may be awaiting a message and not know precisely where to receive it, or they may receive a message, but not know where or to what it is attached. The Proximity-Sensitive System is built on top of existing metadata, such as Bluetooth's device class and name, which allows users some insight into what the message's relationship with the physical world actually is, but much of this must be inferred. This is likely to be easier with GPS signals and RFID sensors, although, for example, users will need to know that their GPS location has been updated and is current. Given that sometimes GPS signals are low or blocked by physical structures, this is not always the case; similarly, it is possible that in some cases, such as multiple RFID sensors stuck in the pages of a book, several sensors may be triggered simultaneously, and it will be hard to determine the precise message-sensor relationships. There is a clear design-relevant point to make about making as much information available to users as to the nature of the signals. Making visible to the user the type of sensors that can be connected to messages or those to which messages are attached, their signal range, how recently the sensor was detected, and making relevant metadata relating to the message sensor available are clearly valuable aides in supporting users to make meaningful interpretations about the message. This point can be extended beyond proximity-sensitive messaging to other proximity-sensitive services. Some of these design considerations are evidently dependent on whether these are multi-sensor systems, but making the nature of the service-sensor relationship transparent is important in determining precisely what, where and how this relationship is embodied.

- The idea that sensors could be 'improved' so that signals are more accurate or could be received in a wider range of environments may not be as useful as it initially appears. The very fact that constraints are imposed on system use as it currently operates may allow users to make better informed judgements on how their message might be interpreted or about what the sender had intended. That a message could not be received and is understood not to be receivable in certain contexts (or vice versa) is important in enriching the meaningfulness of communication, and ensuring uniformally pervasive reception across sensor types may have detrimental communicative value. The utility of this seamfulness (see Chalmers and Galani, 2004) of sensor reception may therefore actually carry value to users, although this would need to be examined in field evaluations. This final point on seamfulness in Proximity-Sensitive Systems is particularly important in that it offers the potential to both form and aid users' understanding about the nature of the proximal connection with the world, and allows its users (as well as designers of future services on such systems) to be creative with the ways that personal communications and other services are provided, by making use of these seams in connectivity to support interpretation. This is in marked contrast to systems like Place Lab, in which all of the sensors in the system are fused into a single notion of location hence not allowing users to make any particular interpretations of *how* those places might have been understood or selected by the sender, to convey their message.

- Just as messages can be associated with people, places and things, so can this information be used for tracking whereabouts of these entities, introducing potential problems in intrusions into personal privacy and the ability to track where people have been and what they have come into contact with. This is not an issue wholly unique to this technology, and has also been considered in pervasive computing applications (Bhaskar and Ahamed, 2007) and location-based technology developments such as Place Lab (Hong *et al.* 2003) and Reno (Consolvo *et al.*, 2005). However, unlike location-based systems, the notion of location within Proximity-Sensitive System Architecture that is inferred from proximity (other than by its GPS component) which adds a degree of fuzziness to a user's actual whereabouts (i.e. not knowing the exact location). Whilst proximity relationships may be equally interesting and useful to nefarious users of the system, the precision as to where users of the system have physically been is less easy to determine. Of course, it may be possible to tell what entities that users have come into contact with from accessing the Message Server, although in practice, as the system is currently designed, metadata about environmental proximity sensors is managed at the level of the local device, and the only information sent to the remote server (and used in the Message Server) are the abstract, unique identifier details, such as RFID codes and Bluetooth MAC addresses, friendly name that are much harder to uniquely identify with a particular person, place or thing.

- It is important to note that the intention with the system under discussion is not to make any more information about people or devices available than is already present: the information being utilised is what users are already making available through, for example, their use of a Bluetooth enabled mobile phone. Users are already publicly broadcasting this information, and this information is simply being as an enabler for determining proximity. Furthermore, in the system that has been developed, there is explicitly no feedback to message senders that their messages have been received (as can be requested in SMS/text messaging for example) so that user activities cannot be traced. In addition, Proximity sensitive messaging provides better privacy compared to paper based messaging as it only makes the information available to the intended recipient, rather than anyone nearby.

In summary, designers can use the Proximity-Sensitive System Architecture as a basis for provisioning a variety of proximity-sensitive services. However, this is not to suggest that it is the only technological solution for building proximity-sensing systems. Rather the intention in developing this Architecture is to stimulate ideas relating to technological convergence and incorporating mobile entities to resolve issues such as ubiquitous coverage, mobility, proximal scalability and communication in relation to any informationally-interesting entities that would otherwise present major challenges to context-awareness in highly mobile and dynamic environments.

6.7 Chapter Conclusion

This chapter has described the evaluation process carried out on the proximity based approach and proof-of-concept prototype. The user study provided information on participants' understanding of proximity, and how they use proximity in their own lives. It elicited users' views on existing messaging systems and their vision for Proximity-Sensitive services.

From a designer or service provider perspective, the participants highlighted a number of important issues relating to security, reliability and privacy which could provide guidance on issues to those keen to build future prototypes. The cost implications discussed may also prove useful to commercial mobile application developers and service providers such as mobile network operators when considering provision of such services to their potential customers. It also discussed some design details from the researcher's point of view to highlight some interesting ideas and drawbacks of the architecture which cannot be drawn from a user study. The next chapter presents the conclusion of this thesis and discusses potential direction for future work.

Chapter 7: Conclusion

7.1 Thesis Summary

This thesis began with the premise that the unique characteristics of proximity could be exploited to provide mobility and scalability to context-aware systems, overcoming some of the limitations of existing systems imposed by their infrastructure dependence and inability to selectively target information delivery thus avoiding information overload. In pursuance of this, the thesis set out to identify suitable technologies for designing such systems. It further explored the possibilities of implementing the design on resource constrained mobile platforms.

Central to the research was the design of a Proximity-Sensitive System Architecture based on which a proof-of-concept prototype was produced. The evaluation of the proof-of-concept was conducted using a purposive sample user study, which helped to identify users' understanding of proximity, their views on and expectations of proximity based communication and their opinion on the proof-of-concept created, especially those features they considered would enhance or add a new dimension to their personal and professional mobile communications.

The development of the proof-of-concept prototype demonstrated that the idea of using proximity and existing technologies was practical, and that similar systems could be developed to support proximity-sensitive services. The user study for its part, revealed that the users sampled had an understanding of the concept of proximity. More importantly, from a design and commercialisation point of view, the study highlighted some of the unmet needs of these users and their concerns regarding costs, privacy and other related issues. The two-fold evaluation clearly demonstrated that the solution offered in this thesis, though not the only possible one, is a workable solution that overcomes existing difficulties. However, the highlighted issues need to be resolved if this design were to be commercialised.

The theoretical analysis that follows reflects on the experiences gained through the course of the research and draws together the contributions it makes to the area of proximity-sensitive services. It also raises some issues from a user point of view that need to be addressed in the hope that it would prove useful to researchers who wish to build on this work or find new directions for enabling context-aware systems to exploit the untapped potential offered by proximity. The next three sections of this chapter therefore present the contribution the research makes to context-aware systems and discusses the limitations of the research before indicating some areas for future work.

7.2 Contributions

The contributions of this thesis are discussed below under five different but closely related areas.

7.2.1 Novel Approach for Supporting Mobility

As a means of supporting device mobility where the devices and the entities with which they interact may be mobile, the thesis examined the concept of spatial proximity and its unique characteristics. The thesis highlights that proximity has a relational property in mediating the relationship between communicating entities, and thus has the capability to offer communication between entities even when their geographical location is not known.

Unlike location-based systems, proximity allows sensing to take place in relation to static (e.g. a place, a building) and mobile (users' mobile devices, books) entities, providing mobility support to systems. This mobility support is based on a proximal relationship that is often represented in the form of 'nearness' to an entity: a person, a place or an object, rather than just location. Thus, proximal relationships can be maintained even when the communicating entities are mobile. There has been a growth of interest in systems that exploit contextual information beyond static sensors. FarCry is one of the systems which looked into supporting mobility through the use of proximity sensing. However, like many other systems, FarCry relies on a particular technology (WLAN) and face-to-face connection. As a result a sender's device in FarCry system has no control over file delivery beyond the immediate vicinity and fails to work outside WLAN coverage areas. Proximity-Sensitive System Architecture discussed in this thesis provides support for mobility through a variety of sensors and able to target information delivery beyond immediate vicinity (i.e. sender has control over who receives the information even when the recipient is not in the immediate vicinity). In addition, Proximity-Sensitive System offers better support for device security as it does not allow information to be copied to all entities connected to the network.

7.2.2 Support for Scalable Interaction

The thesis focused on context-aware interaction, where the interaction is driven by proximal relationships and not by location. Within this notion of proximity-driven interaction, it addressed scalar issues of proximity that are naturally derived from spatial distance. Based on this spatial distance, and taking advantage of technological convergence, the notion of 'nearby' was exploited to offer interaction at three different levels of scale, short, medium and long range. The user study has contributed through verifying the appreciation users have for this type of scalable interaction. Point-to-GeoBlog has looked into scalability to a certain extent. It has discussed scaling in four different levels by zooming in and out. The lowest zoom level allows the user choose close places with high precision while the highest zoom level lets selecting distant places with less accuracy. However, GeoBlog has focused only on scaling areas around locations and has not discussed scaling in relation to mobile entities. Proximity-Sensitive System takes a step further. It covers scaling in relation to locations (or places) and mobile entities by using GPS, ad-hoc sensors with different coverage ranges (e.g. Bluetooth and RFID) respectively.

7.2.3 Identifying and Assessing Candidate Technologies

The research investigated the existing technological landscape and provided an overview on candidate technologies for enabling proximity-sensitive services. This included an examination of the main factors that influence the selection of technologies and describes how some of the existing issues such as mobility and scalability can be addressed using technological convergence. This investigation helped in the selection of technologies to create the system architecture for context gathering, and implement it into a proof-of-concept for enabling a proximity-sensitive personal messaging service. The information thus derived has broader relevance to context-aware, ubiquitous computing and Human Computer Interaction (HCI), where they rely on sensors to gather information about their surrounding environments to provide services in relation to them. The knowledge gained shows how we can utilise the distinctive properties of the various proximity technologies employed to achieve different communicative affordances. In addition, the thesis demonstrates whilst each technology discussed independently has the potential for a variety of applications, they can also be used in concert with one another due to the different capabilities and constraints they carry. This is an important value for systems built on a multi-level platform such as Proximity-Sensitive System as they can exploit technical limitation to system's advantage. For example, RFID's short range characteristic can be utilised to streamline information delivery to a small area.

7.2.4 Ubiquitous Coverage using Existing Technologies

In addition to mobility and scalability, the thesis has examined ways of resolving the coverage problems faced by existing context-aware systems. This is critical for making services available where and when necessary, and for taking communication beyond an area covered by a particular system's infrastructure. The Proximity-Sensitive System Architecture presented in this thesis has provided the facility for ubiquitous coverage through the use of existing sensor infrastructures and ad-hoc sensor networks. Although several systems have attempted to address this coverage problem within context-aware and ubiquitous computing environments, none so far has addressed this issue within the concept of a proximity-based solution or highly mobile and dynamic environments. The work discussed in this thesis has taken advantage of sensor mobility (sensors embedded into mobile entities) and technological convergence to increase the chance of discovery thus enhancing accessibility of services by users across a wide range of environments. Systems such as Place Lab are too focused on resolving coverage issues. However, Place Lab depends on cached beacon locations and thus, provides services only to areas covered by the location database. The Proximity-Sensitive System discussed in the thesis does not require any predefined location information or database.

7.2.5 Identify Users' Understanding, Appreciation and Concerns

The data collected through the user study has helped to understand users' perceptions about proximity and proximity related services, and allows us to assess users' appreciation of the nature and potential of proximity and proximity-sensitive services. The understanding developed through empirical data analysis and interpretation should prove useful to the research community, enabling them to identify what users expect from such systems and tailor services to their needs, rather than forcing them to use what developers believe is best i.e. it helps to ascertain user requirements and concerns at an early stage. The user study has highlighted that Proximity-Sensitive System offers support for context-aware services outside personal messaging and highlights other variables such as time that could help to streamline information delivery.

7.3 Limitations

The thesis focused on a qualitative evaluation to validate the effectiveness of the approach adopted to provide better coverage, mobility and scalability to context-aware systems. In order to do this, the evaluation process created a proof-of-concept prototype and collected user feedback through demonstration. This form of qualitative evaluation is useful to demonstrate the architectural features and gather information from users at an early stage of the development process. It allowed participants to keep an open mind about the situations in which the system could be used and comment freely without having to deal with a constrained situation. However, it could be argued that the interview data is subjective i.e. data collected depends on what users feel and think at that time. In addition, the evaluation process adopted does not provide any detail on the quality of the service.

Another problem with the user study is that it is limited in its ability to test the full potential of such a system in a real life environment. The questions put forward to users were open-ended allowing them to envision such a system in their own lives and discuss its usefulness in real-life situations. It is important to note that the participants were not given the chance to use the system over a long period of time in real-life situations which would have allowed us to collect more accurate and constructive feedback from the participants. This would also have given the opportunity to test the feasibility of the system in real network conditions with fluctuating coverage and network congestion. While this is a limitation of the evaluation method used, performing evaluation in real-life settings would prove extremely difficult without making enhancements to the present prototype, particularly in terms of its robustness. The next section describes some of the areas that need attention.

7.4 Future Work

The work of this thesis was carried out to support its scope of investigation. As a result the thesis focused on proximity sensing rather than connectivity (for routing messages via wireless networks) or interface design for users. However, for reasons of evaluating the usefulness and usability of the system in a real life environment it would be necessary to provide an efficient and reliable wireless connection and user interface. To address these, the following areas warrant further research and development.

7.4.1 Message Routing via Mobile Phone Networks

The messaging routing mechanism that was implemented does not provide ubiquitous coverage outside of a WLAN network. The purpose of implementing the application was to evaluate the underlying architecture that was based on the concept of proximity. It was not the intention to provide message routing for the system. Nevertheless, developing message routing-based on mobile phone networks will be able to provide almost ubiquitous coverage and would be a more suitable option for testing the true potential of such a system architecture.

7.4.2 User Interface Refinement

The user interface element described in the thesis was incorporated to test the architectural and technological features, but not to provide an interface that was suitable for user testing or to create a final product release. Research is needed in this area for designing, evaluating and implementing an appropriate user interface for the proximity-sensitive messaging system so that users are able to understand and make effective use of the communicative resources that such a system can offer to its message senders and recipients.

7.4.3 GPS Map Module Implementation

GPS was incorporated into the system design to provide long range messaging (in relation to a place such as airports, car parks, motorway junctions) and messaging in places where other sensors are not available. Although the Proximity-Sensitive System Architecture has the capability to take input from a geographical map based application, the proof-of-concept has focused on text based input (100m, 500m or 1Km). Implementing a spatial map module into the prototype would allow users to make a better judgement about the area they want to cover. For example, people would be able to define areas for message delivery by simply drawing on the map.

7.4.4 Sensing Technology Adaptation

Chapter 3 provided rationale for choosing a diverse technology based solution for proximity sensing. It also discussed how multiple technology based systems introduce problems in terms of gathering context data (each type of sensor provides different data). In order to resolve this, context gathering program codes were kept separate from the context-aware application code. In addition, each technology had its own set of context gathering code and context file. Although this method (maintaining separate code modules) has given the flexibility to incorporate any different number of technologies into the system without making changes to other software components, the current proof-of-concept only supports Bluetooth, RFID and GPS data. This could be extended to include various other sensors such as WLAN and Barcodes by writing code to gather context information from them. Allowing the system to adapt to more than three technologies will increase the system's coverage even further. Moreover, it may help the prototype to support more than three levels of scalability, and offer different use and communicative affordances in its application.

7.5 Concluding Remarks

Context-aware communication has a long way to go before it becomes a reality. The provision of access to information in relation to people, places and objects is poorly supported in existing mobile and ubiquitous computing technologies. Current context-aware systems and prototypes have primarily focused on location. More specifically, they have relied on the sensors embedded in the environment (i.e. sensors attached to static entities) or geographical coordinates. This causes problems for users when it comes to supporting context-aware communication in relation to mobile entities or deploying context-aware systems (deploying Environmental Sensors to cover the entire earth is not practical). As a step closer to making such systems a reality, this thesis focused on devising a solution to context-aware services based on the concept of proximity. In addition, it exploited technological convergence to provide the scalability and ubiquitous coverage that can be realised on proximity-based systems through the use of existing infrastructures and technologies. Developing Proximity-Sensitive Systems is not a simple matter for a range of technical reasons (e.g. battery limitations, device and infrastructural limitations, vested commercial interests) and interaction reasons (e.g. how to represent sensors to senders and recipients), some of which were encountered during the development of the prototype. Other limitations may emerge from detailed user studies or when steps are taken towards the commercialisation of such Proximity-Sensitive Systems.

The way forward from this thesis is an experimental prototype for field trial to better understand its performance and usability, and investigate the enormous opportunities it offers for novel forms of context-sensitive information access and communication services before implementing a system for real world.

References

Abowd, G. D., Atkeson, C.G., Hong, J., Long, S., Kooper, R. and Pinkerton, M. (1997). 'Cyberguide: A Mobile Context-aware tour guide' *Wireless Networks*, 3 (5) pp.421-433.

Abowd, G. D., Iftode, L. and Mitchell, H. (2005). 'Guest Editors' introduction: The Smart Phone – A First Platform for Pervasive Computing', *IEEE Pervasive Computing.* April-June 4(2). pp. 18-19.

Andronikos, N., O'Connor, A., Abell, G., Clarke, S. and Cahill, V. (2004). 'LATTE: Location and Time Triggered Email' *International Conference on Pervasive Computing and Communications (PCC-2004)*, Las Vegas, USA. 21 – 24 June. CSREA Press, p. 813-819.

Arteaga, M . (2005). *MSDN Forum: Compact Framework 2.0 and FTP*. Available at: http://social.msdn.microsoft.com/forums/enUS/netfxcompact/thread/941951d8-5f3c-46c5-bc9f-b6d8e3fac5fa/ (Accessed 15 October 2007)

Beale, R. (2005). 'Supporting Social Interaction with the Smartphone – Designing Pervasive Systems', *IEEE Pervasive Computing.* 4 (2) pp. 35-41.

Bell, M., Chalmers, M., Barkhuus, L., Hall, M., Sherwood, S., Tennent, P., Brown, B., Rowland, D. and Benford, S. (2006). 'Interweaving Mobile Games with Everyday Life', *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2006)*, ACM Press, pp. 417-426.

Betke, K. (2001). *The NMEA 0183 Protocol.* Available at: http://www.tronico.fi/OH6NT/docs/NMEA0183.pdf (Accessed 10 June 2007)

Beyer, H. and Holtzblatt, K.(1998). *Contextual Design: defining customer-centred systems.* San Francisco, California: Morgan Kaufmann Publishers.

Bhagwat, P. (2001). 'Bluetooth: Technology for Short Range Wireless Apps', *IEEE Internet Computing.* 5(3). Available at: http://www.winlab.rutgers.edu/~pravin/publications/papers/BT-tutorial.pdf (Accessed 15 April 2007)

Bhaskar, P and Ahamed, S. (2007). ‘Privacy in Pervasive Computing and Open Issues’, *Proceedings of the Second IEEE International Conference on Availability, Reliability and Security (ARES 07),* April 10-13, IEEE CS Vienna, Austria, pp. 147-154.

Borriello, G., Chalmers, M., LaMarca, A. and Nixon, P. (2005). ‘Delivering Real-world Ubiquitous Location Systems’, *Communications of the ACM*, March 2005. 48 (3). Pp. 36-41.

Björk, S., Falk, J., Hansson, R. and Ljungstrand, P. (2001). ‘Pirates! - Using the Physical World as a Game Board’, In *Proceedings of the 8th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT),* Tokyo, Japan, 9-13 July.

Blackberry (2008). Available at: http://www.blackberry.com (Accessed 10 June 2008)

Blankenbeckler, D. (2000). *Wireless Developer Network: An Introduction to Bluetooth.* Available at: http://www.wirelessdevnet.com/channels/bluetooth/features/bluetooth.html (Accessed 15 April 2007)

Bluetooth (2007). *Bluetooth Basics.* Available at: http://bluetooth.com/Bluetooth/Technology/Basics.htm. (Accessed: 14 June 2008).

Bluetooth (2008). *Bluetooth: Overview of Operations.* Available at: http://bluetooth.com/Bluetooth/Technology/Works/Overview_of_Operation.htm (Accessed: 19 December 2008)

Bluetooth SIG (2006). *Bluetooth Wireless Technology Surpasses One Billion Devices.* Available at: http://www.bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Wireless_Technology_surpasses_one_billion_devices.htm (Accessed 09 March 2008).

Broadcom (2008). *Broadcom: Bluetooth.* Available at: http://www.broadcom.com/products/Bluetooth (Accessed 13 December 2008).

Brown, P.J. (1998). ‘Triggering information by context’ Springer-Verlag, *Personal Technologies.* 2(1), pp. 1-9.

Brummit, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). ‘Easy living: Technologies for intelligent environments’ *In Symposium on Handheld and Ubiquitous Computing*, Bristol, UK. Springer-Verlag London, pp. 12–27.

BT (2005). *BT SMS Gateway - Frequently Asked Questions*. Available at: http://www.google.co.uk/search?hl=en&q=BT+%282005%29.+BT+SMS+Gateway+-+Frequently+Asked+Questions%23&meta= (Accessed 14 December 2008).

Burge, J and Brown, D. (2000). 'Reasoning with Design Rationale' In Artificial Intelligence in Design 2000, Proceedings of AID'00: the 6th International Conference on AI in Design, (Ed.)

Burge, J. and Brown, D. (2002). Integrating Design Rationale with a Process Model. Workshop on Design Process Modelling, Artificial Intelligence in Design, Cambridge, UK.

Burrell, J. and Gay, G. (2002). 'E-graffitti: evaluating real-world use of a context aware system' *Interacting with Computers*, 14 (4), pp. 301-312.

Burrell, J., Gay, G., Kubo, K. and Farina, N. (2002). 'Context-Aware Computing: A Test Case' *The Fourth International Conference on Ubiquitous Computing* (*UBICOMP02)*, Gothenburg, Sweden. 29 September – 1 October 2002. 2498, pp. 1-15.

Caughey, G. (2004). MSDN: NET Compact Framework and Smart Device Programming: Feature of VS.NET. Available at: http://msdn.microsoft.com/chats/transcripts/mobileembedded/embedded_031104.aspx (Accessed 21 October 2007).

Chalmers, M. and Galani, A. (2004). 'Seamful Interweaving: Heterogeneity in the Theory and Design of Interactive Systems', *In Proceedings of the ACM DIS*, Cambridge, MA, USA, 1-4 August 2004. pp. 243-252.

Cheverst, K., Davis, N., Mitchell, K and Friday, A. (2000). 'Experiences of developing and deploying a context-aware tourist guide: The GUIDE project', In *Proceedings of Conference on Mobile Computing and Networking MOBICOM'2000*, Boston Massachusetts, USA. 6-11 August 2000. New York NY, USA: ACM Press, pp. 20–31.

Chin, T-L., Ramanathan, P., Saluja, K. and Wang, K.C. (2005). 'Exposure for collaborative detection using mobile sensor networks', *In Proceedings of the 2nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'05)*, Washington, DC, November.

Consolvo, S., Smith, I.E., Matthews, T., Lamarca, A., Tabert. J. and Powledge, P. (2005). 'Location disclosure to social relations: why, when & what people want to share'. *In proceedings of ACM CHI 2005*, Portland, Oregon, USA. 2-7 April, 2005. pp. 81-90.

Coulouris, G., Kindberg, T., Schiele, B., Schmidt, A. and Rehman, K. (2005). 'What makes for good application-led research in ubiquitous computing?', *UbiApp Workshop*, Munich, Germany, May.

Dahlberg, P., Ljungberg, F. and Sanneblad, J. (2000). 'Supporting opportunistic communication in mobile settings', *In CHI 2000 Extended Abstracts on Human Factors in Computing Systems*. The Hague, The Netherlands. 1-5 April. New York, NY, USA: ACM Press, pp. 111-112.

Davis B. and Karahalis, K. (2005). 'Telelogs: a social communication space for urban environments', *In proceedings of the 7th International Conference on Human Computer interaction with Mobile Devices & Services (MobileHCI '05)*, Salzburg, Austria. 19-22 September. New York, NY, USA: ACM Press, pp.231-234.

Dey, A. K. (2000). 'Providing Architectural Support for Context-aware Applications', *PhD thesis*. Georgia Institute of Technology. Available at: http://www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf (Accessed 5 July 2008)

Dey, A. K. and Abowd, G.D. (2000). 'Towards a Better Understanding of Context and Context-awareness', *CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness*, New York, USA: ACM Press, pp. 1-6.

Dhawan, C. (1996). *Mobile Computing: System Integrator's Handbook*, New York, USA: McGraw-Hill Inc.

Eagle, N., Pentland, A. (2004). 'Social Serendipity: Proximity. Sensing and Cueing', *MIT Media Lab Tech Note 580*, Available at: http://hd.media.mit.edu (Accessed 27 June 2007)

Esbjörnsson, M. and Brunnberg, L. (2001). 'PlaceMemo: Using GPS and Mobile Computers to Augment the Roads', *Presented at Nordic Interactive*, Copenhagen, Denmark, November 2001.

Esbjörnsson, M. and Juhlin, O. (2002). 'PlaceMemo: Supporting Mobile Articulation in a Vast Working Area through Position Based Information', *In proceedings of ECIS'02*. Available at: http://www.cs.umu.se/education/examina/Rapporter/472.pdf (Accessed 13 January 2009)

Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore. E. and Bylund, M. (2001). 'GeoNotes: Social and navigational aspects of location-based information systems', *In Proceeding Ubicomp 2001: Ubiquitous Computing International Conference*, Atlanta Georgia, USA, 30 September – 2 October, 2201/2001. Berlin: Springer, pp. 2-17.

Falk, J., Ljungstrand, P., Björk, S. and Hansson, R. (2001) 'Pirates: Proximity-Triggered Interaction in a Multi-Player Game', *In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2001)*, Seattle, Washington. 31 March – 5 April. New York NY USA: ACM Press, pp.119-120.

Figueira, J. (2006). 'Enumerating Devices and Services with the Widcomm Bluetooth Stack' Available at: http://www.pocketpcdn.com/print/articles/?&atb.set(a_id)=7800&atb.set(c_id)=78 &atb.perform(details) (Accessed 27 October 2007).

Finan, T. (2003). 'MSDN: Developing Applications for Windows Mobile-Based Smartphones', Available at: http://msdn2.microsoft.com/enus/library/ms838170.aspx (Accessed 19 October 2007).

Foley, M. (2006). Bluetooth SIG: Bluetooth® wireless technology surpasses one billion devices. Available at: http://www.bluetooth.com/Bluetooth/Press/SIG/bluetooth_wireless_technology_surpasses_one_billion_devices.htm (Accessed 14 September, 2008).

Forno, F., Malnati, G. and Portelli, G. (2005). 'Design and Implementation of a Bluetooth ad hoc network for indoor positioning', *Software: IEE Proceedings* October, 2005, 152 (5), pp. 223-228.

Franson (2006). *GpsTools: Develop GPS and Mapping Applications.* Available at: http://franson.com/gpstools/ (Accessed 19 July 2007)

Franson (2007). *Bluetools 1.2*. Available at: http://franson.com/bluetools (Accessed 21 October 2007).

Gellersen, H. W., Schmidt, A. and Beigl, M. (2002). 'Multi-Sensor Context-awareness in Mobile Devices and Smart Artefacts', ACM Jornal Mobile Networks and Applications. October 7(5). pp. 341-351.

Geoframeworks (2008). Available at: http://www.geoframeworks.com (Accessed 14 December 2008).

Getting, I. 'The Global Positioning System', *IEEE Spectrum*, 30(12), December 1993. pp. 36-47.

Gustafsson, A. (2003). 'PlaceMemo: A Prototype of a Context-aware Information System', *Masters Thesis*, Umeå University, Sweden.

Harroud, H., Khedr, M and Karmouch, A. (2004). ‘Building policy-based context aware applications for mobile environments’, Lecture Notes in Computer Science, 3284, Berlin: Springer, pp. 48-61.

Harter, A. and Hopper, A. (1994). ‘A Distributed Location System for the Active Office’, *Network IEEE*, 8(1) January, pp. 62-70.

Harter, A. and Hopper, A. (1997). ‘A New Location Technique for the Active Office’, *IEEE Personal Communications*, 4(5): pp. 43–47, October.

Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P. (1999). ‘The Anatomy of a Context-Aware Application’, *In Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking-MobiCom 99*, Seattle, Washington, USA, 15-20 August. New York NY USA: ACM Press, pp. 59-68.

Holmquist, L.E., Falk J. and Wigström, J. (1999). ‘Supporting group collaboration with inter-personal awareness devices’ *Journal of Personal Technologies*, 3(1-2), pp.13-21.

Hong, C., Kim, H., Cho, J., Cho, H. and Lee, H. (2007). ‘Context Modelling and Reasoning Approach in Context-Aware Middleware for URC System’, *In Proceedings of the World Academy of Science, Engineering and Technology*, Bangkok, Thailand. 14-16 December. 26, pp 129-133.

Hong, J. I., Boriello, G., Landay, J.A., McDonald, D.W., Schilit, B.N. and Tygar, J.D. (2003). ‘Privacy and security in the location-enhanced world wide web’, *In Proceedings of Fifth International Conference on Ubiquitous Computing: Ubicomp 2003* (Workshop on Ubicomp Communities: Privacy as Boundary Negotiations), Seattle WA.

Hong, J, I. and Landay, J, A. (2001). ‘An Infrastructure Approach to Context-Aware Computing’, *Human Computer Interaction*, 16(2), NJ, USA: Erlbaum Associates Inc, pp. 287-303.

Howard, A., Maja, M. and Gauray, S. (2002). ‘Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem’, *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS'02)*, Fukuoka, Japan, 25-27 June 2002.

Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B., Druin, A., Plaisant, C., Beaudouin-Lafon, M, Conversy, S., Evans, H., Hansen, H., Roussel, N., Eiderback, B., Lindquist, S. and Sunblad, Y. (2003). ‘Technology probes:

inspiring design for and with families', In Proceedings of CHI 2003. New York NY USA: ACM Press, pp. 17-24.

Iwatani, Y. (1998). 'Love: Japanese Style', *In Wired News, 11 June 1998*. Available at: http://www.wired.com/culture/lifestyle/news/1998/06/12899 (Accessed 2 September 2007).

Johanson, B., Fox A. and Winograd T. (2002). 'The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms', *IEEE Pervasive Computing Magazine* 1(2), April-June, pp. 67-74.

Jose, R., Moreira, A., Meneses, F. And Coulson, G. (2001). 'An Open Architecture for Developing Mobile Location-based Applications over the Internet', *In 6th IEEE Symposium on Computers and Communications*, Hammamet, Tunisia, 2001, Washington DC, USA: IEEE Computer Society.

Juhlin, O. and Ostergren, M. (2006). 'Time to meet face-to face and device-to-device', *In Proceedings of the 8th Conference on human-computer interaction with Mobile Device and Services*. Helsinki, Finland, 12-15 September. ACM, New York, MobileHCI '06, Vol. 159. pp. 77-80.

Kaasinen, E. (2005). 'User Acceptance of Mobile Services: value, ease of use, trust and ease of adoption', *Tampere University of Technology, Helsinki*. http://www.vtt.fi/inf/pdf/publications/2005/P566.pdf

Kindberg, T. (2002). 'Implementing Physical Hyperlinks Using Ubiquitous Identifier Resolution', Tech Report HPL-2001-95, Hewlett-Packard Laboratories. Available at: http://www.hpl.hp.com/techreports/2001/HPL-2001-95R1.pdf (Accessed 18 December 2008)

Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B. and Spasojevic, M. (2000). 'People, Places, Things: Web Presence for the Real World', *In Proceedings of the 3rd Annual Wireless and Mobile Computer Systems and Applications*, Monterey CA, USA, Washington DC, USA: IEEE Computer Society, pp.19.

Kortuem, G., Kray, C., and Gellersen, H. (2005). 'Sensing and Visualising Spatial Relations of Mobile Devices', *In Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology*. Seattle, WA, USA, 23-26 October, USIST'05. ACM, New York, NY, pp 93-102.

Kostakos, V. (2004). 'A Design Framework for Pervasive Computing Systems', *PhD thesis*. University of Bath.

Kreller, B., Carrega, D., Jeya Shankar, P., Salmon, P., Bottger, S. and Kassing, T. (1998). 'A Mobile Aware City Guide. Application', *In Proceedings of ACTS Mobile Communications,* Summit, Rhodes, Greece, 29 January. pp. 60-65.

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G. and Schilit, B. (2005). 'Place Lab: Device Positioning Using Radio Beacons in the Wild', *In proceedings of 3rd International Conference on Pervasive Computing PERVASIVE 2005*, Munich, Germany, 8-13 May, Berlin, Germany: Springer, pp. 116-133.

Lamming, M. and Flynn, M. (1994). "Forget-Me-Not: Intimate Computing in Support of Human Memory," *Proceedings of FRIEND21: International Symposium on Next Generation Human Interfaces*, Tokyo, Japan, 2-4 February, pp. 125-128.

Lane, G. (2003). 'Urban Tapestries: Wireless networking, public authoring and social knowledge', *Personal and Ubiquitous Computing*, 7(3-4), pp.169-175.

Lee, J. (1997). 'Design Rationale Systems: Understanding the Issues', *IEEE Expert*, 12(3), pp. 78-85.

Long, S., Kooper, R., Abowd, G. D. and Atkeson, C. G. (1996). 'Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study', *In the Proceedings of the 2nd ACM International Conference on Mobile Computing and Networking (MobiCom'96)*, New York, USA, November. New York NY, USA: ACM Press, pp. 97-107.

Ludford, P.J., Frankowski, D., Reily, K., Wilms, K. and Terveen, L. (2007). 'Because I carry my cell phone anyway: functional location-based reminder applications', *In Conference on Human Factors in Computing Systems,* Montréal, Canada, ACM Press, pp. 889-898.

Madhavapeddy, A., Scott, D., Sharp, R. and Upton, E. (2004). 'Using Camera-Phones to Enhance Human–Computer Interaction', *Adjunct Proceedings of the Sixth International Conference on Ubiquitous Computing (UBICOMP 2004).* pp. 1-2.

Madhavapeddy, A., Scott, D. and Sharp, R. (2003). 'Context-Aware Computing with Sound', *Proceedings of the Fifth International Conference on Ubiquitous Computing (UBICOMP 2003).* Vol. 2864 of LNCS, Springer-Verlag.

Marmasse, N. (1999). 'comMotion: Context-aware Communication System', *PhD thesis,* Massachusetts Institute of Technology. Available at:

http://www.media.mit.edu/speech/papers/1999/marmasse_thesis99_commotion.pdf (Accessed 10 January 2008)

Marmasse, N. and Schmandt, C. (2000). 'Location-aware information delivery with comMotion', *In Proceedings of HUC'2000 (Handheld and Ubiquitous Computing)*, Bristol, UK, 25-27 September. Springer-Verlag, pp. 157-171.

Microsoft (2008). MSDN:Bluetooth Protocol Stack. Available at: http://msdn.microsoft.com/en-us/library/ms890946.aspx (Accessed 13 December 2008).

Mitchell, K. (2002). 'Supporting the Development of Mobile Context-aware Systems', *PhD thesis*, Lancaster University, UK. Available at: http://info.comp.lancs.ac.uk/publications/Publication_Documents/theses/KeithMitchell.pdf (Accessed 14 October 2008).

Mobile Bristol (2002-2005). Available at: http://www.mobilebristol.co.uk/flash.html (Accessed 14 August 2007).

MobileInfo (2001). *Wireless LANs*. Available at: http://www.mobileinfo.com/Wireless_LANs/802.11a_802.11b.htm (Accessed 05 July 2008).

MSDN Magazine, (2004). Resource File: Mobile and Embedded Application Development. Available at: http://msdn.microsoft.com/msdnmag/issues/04/02/ResourceFile/default.aspx (Accessed 16 October 2007)

msmobiles(2005). *Review: Using Bluetooth LAN Access Point with Pocket PC*. Available at: http://www.msmobiles.com/catalog/i.php/563.html (Accessed 05 February 2009).

Multaharju, M., Koskinen, K., Spacil, R., Ikonen, J. and Porras, J. (2004). 'Floating Note: a location based messaging application', *In 2nd Workshop on Applications of Wireless Communications*, Lappeenranta, Finland. 5 August, pp. 9-16.

Nam, C. S. and Konomi, S. (2005). 'Usability Evaluation of QueryLens: Implications for Context-aware Information Sharing using RFID', *In Proceedings of the International Conference Human-Computer Interaction*, Phoenix, AZ, USA, ACTA Press, pp.90-95.

Neisse, R., Wegdam, M. and van Sinderen, M. (2006). 'Context-Aware Trust Domains', *In 1st European Conference on Smart Sensing and Context*, Enschede, The Netherlands, 25-27 October, Springer, pp. 234-237.

Nelson, G. J. (1998). ‘Context-aware and location systems’, *PhD dissertation*, University of Cambridge. Available at: http://www.acm.org/sigmobile/MC2R/theses/nelson.ps.gz (Accessed 6 December 2008)

Newby, P. (2006). *Symbian: Operating System Services*. Available at: http://developer.symbian.com/main/documentation/technologies/os_services/index.jsp (Accessed 10 December 2008).

Nokia (2007). *Nokia Sensor*. Available at: http://www.nokia.com/sensor. (Accessed 18 July 2007).

Palm (2008a). *Palm Developer network: Platform Options*. Available online at: https://pdnet.palm.com/wps/portal/pdnet/DEVELOP_GETTING_STARTED (Accessed 10 December 2008).

Palm(2008b), *Smartphones.* Available at: http://www.palm.com/us/products/smartphones (Accessed 10 June 2008)

Pascoe, J. (1998). ‘Adding generic contextual capabilities to wearable computers’, *In Proceedings of the 2nd IEEE International Symposium on Wearable Computers (ISWC'98)*, Pittsburgh, USA, 19-20 October. IEEE, pp. 92-99.

Patrick. (2008). *Phidgets Support and discussion Forum - Error when installing in Windows Mobile 2003*. Available at: http://www.phidgets.com/phpBB2/viewtopic.php?t=2089 (Accessed 16 March 2008)

Perry, M., O'Hara, K., Sellen, A. Harper, R. and Brown, B.A.T. (2001). ‘Dealing with mobility: understanding access anytime, anywhere’, *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8 (4), New York, USA: ACM Press, pp.323-347.

Persson, P., Espinoza, F., Sandin, A., and Coster, R. (2002). ‘GeoNotes: A Location-based Information System for Public Spaces’, *In Proceedings of Mobile HCI 2002*, Pisa, Italy. Berlin, Germany: Springer-Verlag, pp.151-173.

Persson, P and Fagerberg, P (2002). **‘**GeoNotes: a real-use study of a public location-aware community system’, SICS Technical Report. ISSN 1100-3154.

Patrick. (2008). *Phidgets Support and discussion Forum - Error when installing in Windows Mobile 2003*. Available at: http://www.phidgets.com/phpBB2/viewtopic.php?t=2089 (Accessed 16 March 2008)

Priyantha, N., Chakraborty, A. and Balakrishnan, H. (2000). 'The Cricket Location-Support System', *In Proceedings of the 6th International Conference on mobile computing and networks (MOBICOM)*, Boston, MA, 6-11 August. ACM Press, pp. 32–43.

Priyantha, N.B., Miu, A.K.L., Balakrishnan, H. and Teller, S. (2001). 'The Cricket Compass for Context-aware Mobile Applications', *In Proceedings of the 7th International Conference on mobile computing and networks (MOBICOM)*, Rome, Italy, 16-21 July. New York, USA: ACM Press, pp. 1-14.

Rachovides, D. and Perry, M. (2006). 'HCI Research in the Home: Lessons for Empirical Research and Technology Development within the Home'. 20th BCS HCI Conference. (2) pp 30-33.

Rahmani, A. T., Rafe, V., Sedighian, S. and Abbaspour. A. (2006). 'An MDA-Based Modeling and Design of Service Oriented Architecture', *In Proceedings of the Sixth International Conference on Computational Science*. ICCS '06. Vol. 3993, pp. 578-585.

Randell, C., MacColl, I., Muller, H. and Rogers. Y. *(2002). 'Exploring the Potential of Ultrasonic Position Measurement as a Research Tool', IEE Communication Networks and Services Professional Network, editors, First European Workshop on Location Based Services*, Savoy Place, London, September 2002. pp. 171-174.

Rantanen, M., Oulasvirta, A., Blom, J., Tiitta, S., and Mäntylä, M. (2004). 'InfoRadar: Group and public messaging in mobile context', *In Proceedings of NordiCHI 2004*, New York. USA: ACM Press, pp. 131-140.

Redström, J., Dahlberg, P., Ljungstrand, P. and Holmquist, L. E. (1999). 'Designing for Local Interaction'. In: Nixon, P., Lacey, G. & Dobson, S. (Eds.) *Managing Interactions in Smart Environments: 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, Dublin, December 1999, Springer-Verlag, pp. 227-238.

RFID centre. *Introduction to RFID*. Available at: http://www.rfidc.com/docs/introductiontorfid.htm (Accessed 09 November 2007)

RFID Journal (2004). Nokia Unveils RFID Phone Reader. http://www.rfidjournal.com/article/view/834/1/1 (Accessed 28 September 2007)

RFID Journal (2008). Available at: http://www.rfidjournal.com (Accessed 05 December 2008).

Rico, F., Forga, J., Sanvicente, E. and de la Cruz, L. (2006). 'CerTicket Solution: Safe Home-Ticketing Through Internet', *E-Commerce and Web Technologies Book Series*. 4082/2006, Springer, pp 152-161.

Riva,O. (2007). 'Middleware for Mobile Sensing Applications in Urban Environments', *PhD Thesis*. University of Helsinki, Finland.

Riva, O and Toivonen, S. (2007). 'The DYNAMOS Approach to Context-Aware Service Provisioning in Mobile Environments', *Elsevier Journal of Systems and Software*, December 80(12), pp.1956-1972.

Riva, O., Nadeem, T., Borcea, C., and Iftode, L. (2007). 'Context-aware Migratory Services in Ad Hoc Networks', *IEEE Transactions on Mobile Computing*, December 6(12), pp.1313-1328.

Robinson, S., Eslambolchilar, P. and Jones, M. (2008). 'Point-to-GeoBlog: Gestures and Sensors to support User Generated Content Creation', *ACM International Conference Proceeding Series, Proceedings of the 10th international conference on Human computer interaction with mobile devices and services***.** Amsterdam, Netherlands. 2-5 September. New York, USA: ACM Press, pp.197-206.

Roof, L. (2002). MSDN: Getting Started with Visual Studio .NET and the Microsoft .NET Compact Framework Available at: http://msdn2.microsoft.com/en-gb/library/aa446534.aspx. (Accessed 10 October 2007).

Satoh, I. (2007). 'A Location Model for Smart Environments', *Pervasive and Mobile Computing*, March 3(2), Science Direct, pp. 158-179.

Schilit, B., Theimer, M. (1994). 'Disseminating Active Map Information to Mobile Hosts', *IEEE Network*, 8(5), pp.22-32.

Schmandt, C., Marmasse, N., Marti, S., Sawhney, N. and Wheeler, S. (2000). 'Everywhere messaging', *IBM Systems Journal*, 39(3-4), pp.660-677.

Schmidt, A. (2002). 'Ubiquitous Computing – Computing in Context', *PhD Thesis*, Lancaster University, UK.

Schmidt, A., Beigl, M. and Gellersen, H. (1998). 'There is more to context than location', *In the Interactive Applications of Mobile Computing (IMC '98*), Rostock, Germany. 24 -25 November, pp. 893-902.

Schmidt, T. and Townsend, A. (2003). 'Why Wi-Fi Wants To Be Free', Communications of the ACM, 46 (5), New York, USA: ACM Press, pp. 47-52.

Schulzrinne, H., Moghadam, A., Shacham, R., Srinivasan, S., Wu, X. (2007). 'The Vision and Reality of Ubiquitous Computing', *The 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. Philadelphia, USA. 6-10 August. Available at: http://www.cs.columbia.edu/~hgs/papers/2007/mobiquitous-keynote.ppt. (Accessed 14 December 2008)

ScientificComponent (2006). *GPS TOOLKIT .NET*. Available at: http://www.scientificcomponent.com/gps_toolkit_net_mobile.htm (Accessed 19 June 2006).

Seppänen, A. (2002). 'Gathering and using location information in WLAN', *Master's Thesis*, Lappeenranta University of Technology, Finland.

Skove, Mikael, B. and Hoegh, Rune, Th. (2006). 'Supporting information access in a hospital ward by a context-aware mobile electronic patient record', *Personal and Ubiquitous Computing Journal*, 10(4), London, UK: Springer-Verlang, pp. 205-214.

Socket (2007). *Socket Compact Flash RFID Readers - RFID Reader Card 6E RFID.* Available at: http://www.socketmobile.com/products/bar-code-scanning-data-collection/series6/ (Accessed 19 October 2007).

Sun(2009), Java: JSR 256 (\Mobile Sensor API). Available at: http://jcp.org/en/jsr/detail?id=256 (Accessed 10 October 2007)

Tennent, P., Hall, M., Brown, B., Chalmers, M., and Sherwood, S. (2005). 'Three Applications for Mobile Epidemic Algorithms', *In Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services*. Salzburg, Austria, 19-22 September. MobileHCI '05, Vol. 111. ACM, New York, pp. 223-226.

Trochim, W M (2006). *The Research Methods Knowledge Base, 2nd Edition*, Available at: http://www.socialresearchmethods.net/kb (Accessed 01 November 2008).

Ward, A., Jones, A. and Hopper, A. (1997), 'A New Location Technique for the Active Office', *IEEE Personal Communications*, 4(5), pp.42-47.

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992). 'The Active Badge Location System', *ACM Transactions on Information Systems*, 10(1), pp. 91-102.

Want, R., Schilit, B., Adams, N., Gold, R., Petersen, K., Goldberg, D., Ellis, J. and Weiser, M. (1997). 'The Parc Tab Ubiquitous Computing Experiment', *Mobile Computing*, Kluwer Publishing, pp. 45-101.

Weber, G. (2005). Siemens Corporate Web Site featuring Digital Graffiti Service Available at: http://www.infosyncworld.com/news/n/5767.html (Accessed 28 August 2007).

Weiser, M. (1991). 'The Computer for the 21st Century', *In Scientific American*, 265(3), pp. 94-104.

Weiser, M. and Brown, J. (1996). The coming age of calm technology. *PowerGrid Journal*, 1.01 , July. Available at: http://powergrid.electriciti.com (Accessed 7 March 2007).

Yun-Maw, C., Wai, Y., Tzu-Chuan, C. (2005). 'Life is Sharable: blogging life experience with RFID embedded mobile phones', *In Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*. New York, USA: ACM Press, pp. 295-298.

Zyda, M. (2007). 'Educating the next generation of game developers', *In proceedings of IEEE International Conference on Portable Information Devices.* Orlando, USA, 25-27 May. IEEE, pp. 30-34.

3GPP (2004). '3rd Generation Partnership Project (3GPP)', *Technical Specification Group (TSG) RAN; Working Group 2 (WG2); Report on Location Services (LCS)*. Available at: http://www.3gpp.org/ftp/Specs/1999-04/25_series/ (Accessed 14 December 2008).

Appendix A

Implementing the Interface Component: Messenger

The Messenger component relies heavily on the information provided by the Linker. This component was implemented using Visual Studio 2005 Form Designer. The context information such as number of sensors identified (i.e. counts) and their details are (i.e. arrays for Bluetooth, RFID and GPS) passed on to Messenger for updating the Short, Medium, Long radio buttons. These three radio buttons stay disabled until the mobile device finds appropriate sensors or its location in the environment. The high-level implementation of this component is described below.

```
Messenger ()

Load Screen
Refresh Screen based on Discovery results

If RFID count is NOT = 0
       Enable Short Radio Button
Endif

If Bluetooth count is NOT = 0
       Enable Medium Radio Button
Endif

If GPS count is NOT = 0
       Enable Long Radio Button
Endif

Check  messages  in  /toread/  folder  and
refresh Received Messages

Send message to /tosend/ folder when Send
button is clicked
```

Figure A.1: High-Level Implementation for Messenger

The Messenger also provides support for message authoring, gathers information relating to To, Range, Tag to and Message parameters, creates message files and stores them in /toread/ folder as described in Chapter 4 through the code generated within Form Designer and its Objects (e.g. buttons, lists). In addition, it displays messages related to the current context in the ‘Received Messages’ section.

Appendix B

Implementing the Routing Component

Although, implementation of this component is immaterial for this thesis, it was necessary for demonstrating that the context information gathered by Bluetooth, RFID and GPS sensors can trigger context sensitive, or more precisely, proximity-sensitive message delivery. There is no way of demonstrating that the message delivery is triggered based on the context condition, without being able to route the message to the recipient's device. However, quite a few systems (e.g. Davis *et al*., 2005) have already used W-LAN or mobile phone networks to route their messages and it is not the intention to reinvent the wheel. Therefore, the decision made in Chapter 4 is used to implement a simple Routing component: a central http message server, WLAN connectivity and a Router. This implementation is described below.

Http Message Server

The University Server is utilised for storing message files, created an http path (http://people.brunel.ac.uk/~cspgcsu/ProxMS-Clients/) and allowed Clients (senders' and recipients' devices) to access this area through 802.11b on their devices

WLAN Connectivity

Ideal situation is for the senders' and recipients' devices to have ubiquitous connectivity to the http message server, to upload and download messages routinely. For this to happen with WLAN technology, the Clients (senders and recipients) must be within the vicinity of WLAN access points. However, for proof-of-concept, providing ubiquitous connectivity was not important as long as message files are routed to recipients before their devices discover the marked entities (i.e. meet the context condition). The 802.11b technology on the PDAs was considered pervasive enough to support this.

Router

The Router is a software application stored on mobile devices (Clients) that periodically connects to the http message server via WLAN and transfers message files to and from PDAs. It issues a connection request and waits for the server to respond. If the request was successful, it reads the message files and writes those files to the /toread/ directory on the local device. In the same way, the Router must be able to read files in the /tosend/ folder and write them to the message server. However, this uploading functionality is not supported by earlier versions of pocket PC operating systems. Thus, was carried out manually.

Appendix C

Environmental Sensors

As discussed in Chapter 5, the Environmental Sensors for the Proximity-Sensing component is provided by the existing static and mobile Bluetooth sensors and GPS satellites. In addition to these existing sensors, RFID sensors are attached to objects where and when necessary to provide the coverage for delivering messages in relation to them. Figure C.1 illustrates some of the Environmental Sensors that could be used by the Proof-of-concept.

Figure C.1: Environmental Sensors

Appendix D

Questions Used at the Interview

1. Proximity

   Your perception, understanding or interpretation of the word 'proximity'

2. Proximity and Location

   Would you consider these two to be same, similar or different?

   Provide reasons and explain situations to describe their use

3. What type of communication services or medium do you use regularly?

   This question is not restricted to technology based medium. Therefore, please feel free to discuss any medium (e.g. paper, white boards).

   At Work:

   For Personal and Social:

4. Are you happy with the services or mediums you discussed in Question 3? Do they match up to your requirements or expectations?

   Please explain the Likes and Dislikes for each service or medium discussed in Question 3.

5. Have you used Traffic Forecast or Local Yellow Pages on the mobile phone

   If Yes

   Please explain the Likes and Dislikes for each service

Would you like to change or add anything to these services?

If Yes:

Please explain

6. We all meet people and come across objects in different settings. Have you ever been in a situation where you wanted to use someone or an object (e.g. fridge, desk) to pass messages to others?

   For example consider the two situations below

   Situation 1: Sam and Kate are your friends, and you want Sam to pass a message to Kate when they meet next time.

   Situation 2: Leave a message to Kate on her desk.

   Can you provide examples of such situations?

7. Introduce the proof-of-concept for Proximity-Sensitive Messaging System

   Demonstrate and explain that it is based on the proximity between two entities, and explicitly specify that these two entities can meet anywhere and trigger information delivery.

   Discuss the scalability feature and demonstrate it to the participant.

   Can you provide some situations where scalability would be useful?

8. Can you see yourself using such a system in the future?

   If Yes:

   Why would you like to use such a system?

   Where would you want to leave your messages?

   Where would you want to receive your messages?

   If No:

Would you like to explain why such a system is not suitable for you?

How would you prefer to communicate?

Why?

Move to Question 12

9. Encourage participants to think about scaling areas for message delivery. If necessary give them an example to stimulate ideas.

   Would you consider various levels of specificity as a useful feature?

   If Yes:

   Why?

   Where would you use it?

   If No:

   Why?

10. Do you have any concerns or issues about the system?

    If Yes:

    Please explain

11. Can you think of another system that allows access to proximity-sensitive services?

12. Would you like to change anything in the system to make it more appropriate for your use?

    Please explain your answer.

Appendix E
Demonstrating and Describing the Proof-of-Concept

The proof-of-concept was used to describe and demonstrate Proximity-Sensitive System Architecture's role in supporting proximity-sensitive services, particularly proximity-sensitive personal messaging services. All participants had the opportunity learn how the proof-of-concept helps to find and mark entities that are informationally-interesting to them, and later how messages can be tagged to those marked entities.

The demonstration was carried out using two PDAs and a mobile phone. Both PDAs: sender's (say his name is Rob) iPAQ h5550 and recipient's (say John) iPAQ rx 3715 come with built-in Bluetooth and W-LAN sensors. In addition, the proximity-sensitive messaging application was deployed on these two PDAs. The mobile entity (say Kate's mobile phone) Nokia 6600 mobile phone also comes with built-in Bluetooth sensor, but does not require proximity-sensitive messaging application as Kate's device does not actively take part in the messaging between Rob's device and John's device. Before the demonstration the participants were given an overview for the proof-of-concept and interface on the mobile devices. Following this, the process of sending and receiving messages in different levels of spatial specificity was explained.

Sending Messages

It was demonstrated to the participants that the proof-of-concept prototype finds entities in the current environment and enables the radio buttons (short, medium, long) accordingly i.e. if it finds entities for short range it enables the Short radio button. It was also described that the users can choose one of the enabled ranges for leaving their messages. For example, if the user is interested in leaving a medium range message on Kate's mobile device, and if the medium range radio button is enabled the user can tap on the Medium radio button. Once the user has chosen the Medium range button a list of entities that are available for leaving a medium range message is displayed in the 'Tag To:' combo box. During the demonstration the participants were asked to turn on their mobile devices and enable their Bluetooth sensors so that they can see their devices listed in the 'Tag to:' combo box. However, some the participants devices were listed as 'My device' or 'mine' as they have not provided a meaningful friendly name for their devices. For the purpose of demonstration Kate's mobile phone or a participant's mobile device was chosen for leaving the message. Figure E.1 illustrates an example where Kate's mob was chosen as mobile entity for tagging the message. Then John was chosen as message recipient from the contact list. This allowed to type in the message in the text box below the 'Tag To:' combo box. Once all the information is entered the message was sent by tapping on the Send button at the top.

Figure E.1: Medium Range Messaging Screen

It was also explained that the user is free to choose Short or Long range messaging if there was coverage in users' current environment. The screen for these two type of messaging were also shown to the participants and they were explained that short and long range 'Tag To' combo boxes provide a Tag ID (see Figure E.2) and three different long range levels e.g. 100m, 500m 1Km (see Figure E.3) respectively. Once the process of message sending was described and demonstrated it was explained how this message is made available to the recipient.

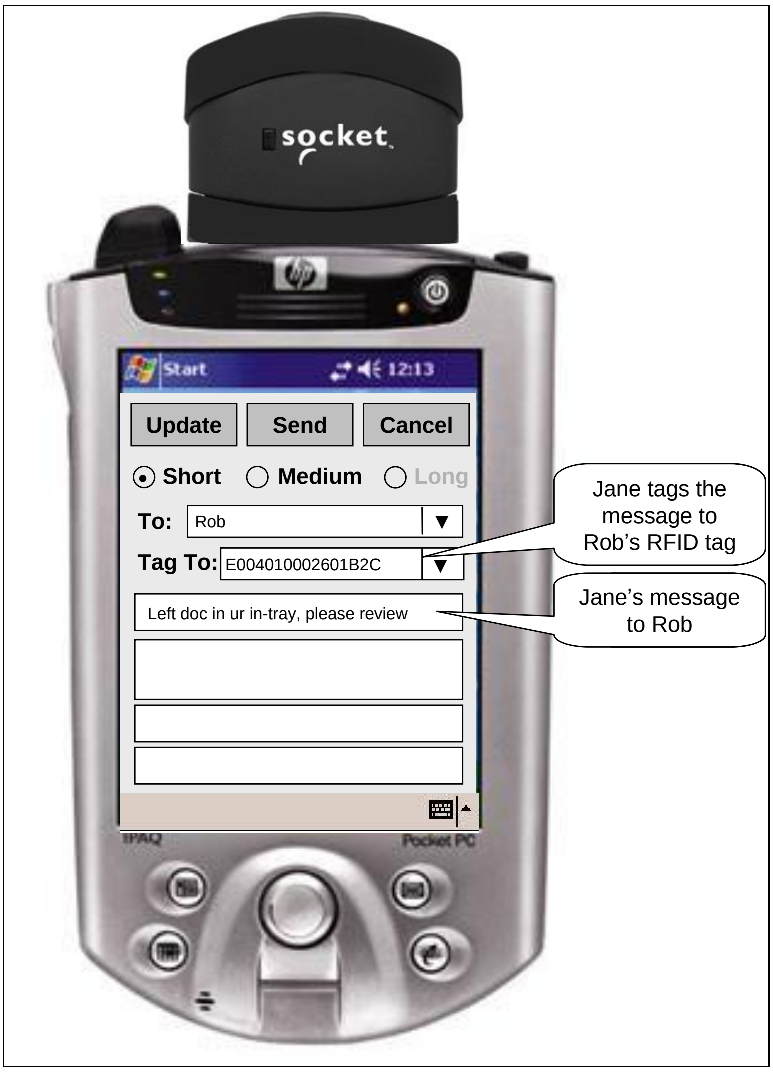

Figure E.2: Short Range Messaging Screen

Figure E.3: Long Range Messaging Screen

Receiving Messages

It was demonstrated to the participants that Rob's message to John is only available when Kate's mobile phone is discovered by John's device. Until then the message is not displayed in the received message box.

E.4: Rob Receives John's Message

When John's device comes within close proximity (close enough to be discovered) then John proximity-sensitive messaging interface lists Rob's message in the received message box. The recipient can highlight and tap on the message to view the message i.e. when the user taps on the message tagged to Kate's mob, the message and sender's details are displayed in the text boxes below the received message box. This is illustrated in Figure E.4. In this particular example (when the message tagged to Kate's mobile device) the triggering of message notification for John's device depends on the proximity range between him and Kate (i.e. when John's Bluetooth sensor can discover Kate's Bluetooth sensor) but not on the geographical location or place. This was demonstrated by delivering messages in various locations.

Following on from this, all participants were shown how short, medium or long range message delivery can be triggered in relation to static entities or places of their choice. It was brought to their attention that short range messaging is triggered when the recipient is very close to the marked tag. They were shown that this may some time require the devices to be very close to the tag, most likely in a deliberate act of swiping over it. In long range messaging, the device has to be physically within the messaging range defined by the sender.

This particular scenario helped to demonstrate that the prototype has the potential to proximate in various levels of scale: narrowing messaging to a proximal range that is most suitable for the messaging situation. In addition, this scenario shows that proximity sensing can take place not only in relation to mobile entities (PDA) but also in relation to fixed entities (a table, a door). The participants were allowed to stick RFID tags on to entities of their choice and discover them using the PDA. This gave us the opportunity to explain how simple it is to tag entities of their choice and later find them to leave messages on those entities.