DOCTOR OF ENGINEERING (ENGD)

Sensing and Interactive Intelligence in Mobile Context Aware Systems

Lovett, Tom

Award date:
2013

Awarding institution:
University of Bath

Link to publication

Sensing and Interactive Intelligence in Mobile Context Aware Systems

submitted by

Thomas Lovett

for the degree of Doctor of Engineering

of the

University of Bath

Department of Computer Science

December 2012

COPYRIGHT

Signature of Author ....

Thomas Lovett

Acknowledgements

I can only imagine how these last few years might have unfolded without the support of various people.

Firstly, my utmost thanks to my supervisor, Eamonn O'Neill, who has counselled, mentored, guided and supported me with interminable patience throughout. Thanks for all your time and effort over the years.

Thanks also to my academic cohort at the University of Bath's Department of Computer Science. In particular: Simon Jones, Andy Ridge, Andrew Chinery, Michael Wright, Vicky Shipp, Laura Benton, Daniel Gooch and Jim Grimmett. Not forgetting those in my EngD cohort and the Systems Engineering Doctorate Centre SEDC either.

Particular thanks to Elizabeth Gabe-Thomas at the University of Bath for her help with viva preparation.

On the industrial side, my thanks to those who helped me and provided the necessary resources for my studies and trips during my time at Vodafone Group R&D, namely: David Pollington (now at GSMA), James Irwin, Kevin Scarr, Roy Davies and Narseo Vallina Rodriguez (now at the University of Cambridge).

I'm further grateful to all my diligent study participants for their role in this work, and thanks to my old friends for their company: Becky Kingdon, Andrew 'Sock' Lock, David Andrews and John Reeks. I'm indebted to Sock for his proof reading services.

To my family, especially my parents, for your irreplaceable presence in my life.

Finally, to Sophie, for your immutable and constant belief in me, regardless of my often capricious outlook.

Abstract

The ever increasing capabilities of mobile devices such as smartphones and their ubiquity in daily life has resulted in a large and interesting body of research into context awareness – the 'awareness of a situation' – and how it could make people's lives easier. There are, however, difficulties involved in realising and implementing context aware systems in the real world; particularly in a mobile environment.

To address these difficulties, this dissertation tackles the broad problem of designing and implementing mobile context aware systems in the field. Spanning the fields of Artificial Intelligence (AI) and Human Computer Interaction (HCI), the problem is broken down and scoped into two key areas: context sensing and interactive intelligence. Using a simple design model, the dissertation makes a series of contributions within each area in order to improve the knowledge of mobile context aware systems engineering.

At the sensing level, we review mobile sensing capabilities and use a case study to show that the everyday calendar is a noisy 'sensor' of context. We also show that its 'signal', i.e. useful context, can be extracted using logical data fusion with context supplied by mobile devices.

For interactive intelligence, there are two fundamental components: the intelligence, which is concerned with context inference and machine learning; and the interaction, which is concerned with user interaction. For the intelligence component, we use the case of semantic place awareness to address the problems of real time context inference and learning on mobile devices. We show that raw device motion – a common metric used in activity recognition research – is a poor indicator of *transition* between semantically meaningful places, but real time transition detection performance can be improved with the application of basic machine learning and time series processing techniques. We also develop a context inference and learning algorithm that incorporates user feedback into the inference process – a form of active machine learning. We compare various implementations of the algorithm for the semantic place awareness use case, and observe its performance using a simulation study of user feedback.

For the interaction component, we study various approaches for eliciting user feedback in the field. We deploy the mobile semantic place awareness system in the field and show how different elicitation approaches affect user feedback behaviour. Moreover, we report on the user experience of interacting with the intelligent system and show how performance in the field compares with the earlier simulation. We also analyse the resource usage of the system and report on the use of a simple SMS place awareness application that uses our system.

The dissertation presents original research on key components for designing and implementing mobile context aware systems, and contributes new knowledge to the field of mobile context awareness.

Preface

Work from this thesis both appears in and relates to the following peer-reviewed publications:

- T. Lovett and E. O'Neill. Simulating User Intervention for Interactive Semantic Place Recognition with Mobile Devices. In Proc. LocalPeMA '12 at RecSys '12, pages 13–18. ACM. 2012. DOI: 10.1145/2365946.2365950
- T. Lovett and E. O'Neill. Capturing Transitions Between Users' Semantically Meaningful Places Using Mobile Devices. In Proc. MCSS '12 at MobiSys '12, pages 11–16. ACM. 2012. DOI: 10.1145/2307863.2307868
- T. Lovett and E. O'Neill, editors. Mobile Context Awareness. Springer Publishing Company, Inc., 2012. ISBN: 0857296248, 9780857296245. DOI: 10.1007/978-0-85729-625-2
- T. Lovett and E. O'Neill. Context Transitions: User Identification and Comparison of Mobile Device Motion Data. In Proc. Workshops at AAAI '11, pages 42–47. AAAI. 2011.
- T. Lovett, E. O'Neill, J. Irwin, and D. Pollington. The Calendar as a Sensor: Analysis and Improvement using Data Fusion with Social Networks and Location. In Proc. Ubicomp '10, pages 3–12. ACM. 2010. DOI: 10.1145/1864349.1864352
- T. Lovett, E. O'Neill, J. Irwin, and D. Pollington. Event-based Mobile Social Network Services. In CAM3SN at MobileHCI '09. 2009.
- T. Lovett and E. O'Neill. Mobile Context-Awareness: Capabilities, Challenges and Applications. In Adjunct Proc. Ubicomp '10, pages 539–540. ACM. 2010. DOI: 10.1145/1864431.1864508
- D. Ferreira, E. Miluzzo, J. Hakkila, T. Lovett and V. Kostakos. Ubiquitous Mobile Instrumentation (UbiMI). In Proc. Ubicomp '12, pages 715–716, 2012.

Additionally, two patent applications have arisen from this work:

- UK Patent Application GB1205162.9: State Transition. Vodafone IP Licensing Ltd. 2012.
- UK Patent Application GB1205160.3: State/Context Detection. Vodafone IP Licensing Ltd. 2012.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

> "*Quis, quid, quando, ubi, cur, quem ad modum, quibus adminiculis*"
>
> (Who, what, when, where, why, in what way, by what means.)
>
> — Hermagoras of Temnos, *c.* 1 B.C.

The preceding quote refers to the 'elements of circumstance' [194], a philosophy that is rooted in a set of interrogatory questions, each of which may be used to elicit a description of a circumstance or event. These elements are perhaps more recognisable in their modern form: the Five Ws (or sometimes 5W1H, for the How question); an informal information gathering method used predominantly by journalists to report on events. The questions ask:

1. **Who** is involved in the event?
2. **What** is the event about?
3. **Where** is it occurring?
4. **When** is it occurring?
5. **Why** is it occurring?
6. **How** is it occurring?

A truthful answer to each question should provide a complete, objective description of the event for good reportage; it should describe the circumstances or *context* of the event. Loosely speaking, context is an informative description of a situation or event, and to take something 'out of context' is to remove relevant information that may affect the interpretation of that something. The popular phrase "context is everything" refers to additional meaning that context can provide beyond what is stated, or the dependence upon context for the true understanding of an event or communication, e.g. in speech and writing, where additional information about what is said or written may be communicated through knowledge of verbal context.

According to the International Telecommunication Union (ITU), nearly 87% of the world's population have access to a mobile device [1]. Many of these devices are smartphones or tablets, and they are often equipped with rich and dynamic user interfaces, as well as a range of hardware sensors. These features, coupled with the sheer popularity and ubiquity of the mobile device, have led researchers to explore ways in which such devices could be used to improve people's everyday lives.

Mark Weiser's 'vision' of ubiquitous computing [236] – which was, arguably, the genesis of the ubiquitous computing (UbiComp) community in computer science – is frequently cited as the idealistic goal for designers of mobile and pervasive computing systems. Free from the constraints imposed by wires, bulky screens, immobile desktops and industrial scale machinery, designers and researchers have questioned the extent to which mobility can enable technology that is truly "indistinguishable" from the "fabric of everyday life". Realising this vision of mobile ubiquitous computing involves the development and integration of work from two primary areas of computer science: artificial intelligence (AI) and human-computer interaction (HCI).

AI is a broad and varied field in which the overarching aim is the creation of intelligent machines (see Figure 1.2). The various AI communities' development of approaches to automated data sensing, statistical inference and machine learning have allowed researchers in the UbiComp community to implement machine intelligence on everyday mobile devices. This is done with a view to making device users' lives easier by offloading burdensome tasks such as search or navigation onto machine intelligence. Ideally, the machine intelligence should perform its tasks perfectly and without need for human supervision. Realistically, however, machine intelligence is likely to make mistakes or fail entirely, which – if the intelligence is not designed well – can result in user annoyance, frustration and general dissatisfaction. Prudent designers should account for this and, in doing so, should design for human interaction with the machine intelligence. HCI is the general study of people's interaction with computers, and HCI researchers work to improve the processes involved in this interaction, as well as people's experiences of the interaction itself. UbiComp researchers have drawn extensively on work from the HCI community during the design and implementation of user interfaces and interaction modes for mobile devices.

Coupling these fields together, one of the today's research challenges is the effective

[1] http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf (Accessed 2012-11-14.)

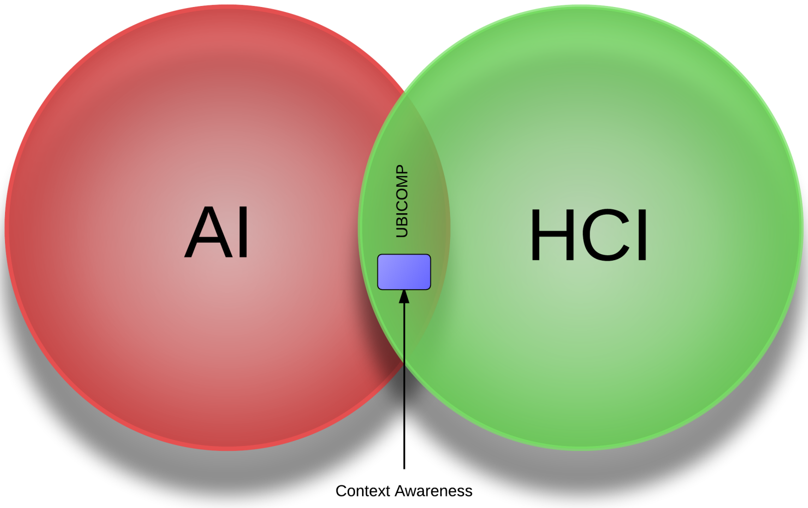


Figure 1.1: *Context awareness scope: context awareness is a sub-field of UbiComp which, in turn, lies in the intersection of AI and HCI.*

and efficient design of *interactive intelligent systems* (IIS) – intelligent systems that users interact with [105] – which comprise of an AI component and an HCI component, and the potentially complex interactions between the two.

So where does context fit in? To give a semi-formal scope: if we imagine the fields of AI and HCI as sets in a Venn diagram (see Figure 1.1), then UbiComp lies in their intersection. Context forms the basis of a popular sub-field (or subset in the analogy) in UbiComp: *context awareness*. A computer is said to be 'context aware' if it can adapt to a given situation [202] or provide relevant information and/or services to a user [57].

Although a popular field within academia, context awareness has recently appeared in commercial products and applications. Apple's Siri system [2] and Google's Now platform [3] are perhaps the most notable modern examples. What is noticeable about these and many other applications is the fact that they operate primarily on *mobile devices*; taking advantage not only of increasing mobile device ubiquity and capability, but also of mobile devices' integration into their users' everyday lives.

This ubiquity of mobile devices, coupled with their ever increasing capabilities, has resulted in a further branch of context awareness in which device mobility plays a

[2] `http://www.apple.com/uk/ios/siri/` (Accessed 2012-11-16)
[3] `http://www.google.com/landing/now/` (Accessed 2012-11-16)

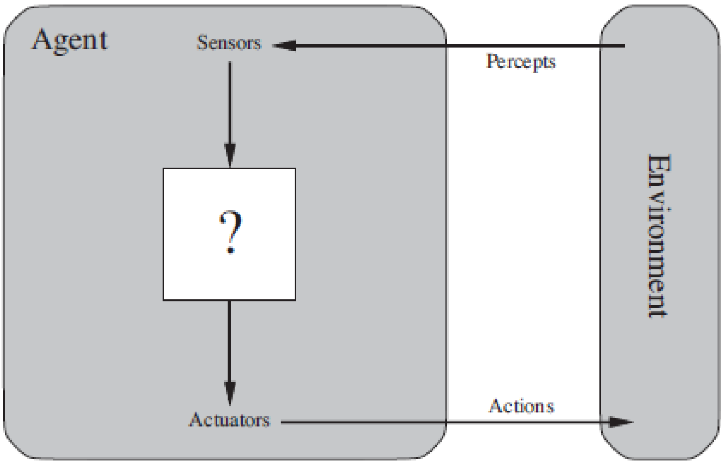


Figure 1.2: *The architecture of an intelligent agent; the core design principle in AI. (Image taken from [199].)*

fundamental role. This is *mobile context awareness*. Mobile context awareness is a fast-moving research field, driven in part by the rapidity of commercial device improvement, and mobile context awareness research has generated valuable work which, in turn, has fed back into industry. However, even in such an evolving field, there are still many issues and problems that have yet to be solved.

In this dissertation, we address the following broad research question:

- $\mathbf{RQ_H}$: How can we improve the design and implementation of mobile context aware systems?

By focusing on a set of questions within mobile context awareness, this dissertation presents a series of contributions that advance the state of knowledge on the design and implementation of mobile context aware systems. The following section outlines the scope of this dissertation, and the succeeding section summarises our contributions in greater detail.

1.1 Dissertation Scope

In this section we define the scope of this dissertation by outlining a set of high level, ongoing research areas in the field of mobile context awareness. Although Chapter 2 covers these in greater detail, we introduce them here to provide a basis for our contribution summary in the next section.

1.1.1 Context Sensing

Although the technical capabilities of mobile devices are generally improving, their functionality is still relatively limited when compared with desktop devices: user interfaces are typically small; resources – particularly power resources – are limited and often constrained; on-device components and sensors are designed for a narrow and specific range of basic functions, e.g. accelerometers for screen orientation changes; and connectivity is highly dependent on device location.

One of the basic requirements for any context aware system is the acquisition of context data. This is undertaken using a set of sensors that translate data from context sources, e.g. people, environments or other devices, into machine-readable data for higher level context inference and learning processes to use. The question here relates to context data sensing: *from where can we source context data, and how might we sense it?* This is an important problem, as the acquisition of context data is one of the most critical aspects of a context aware system. Failure to sense data can dramatically impact on context awareness functionality and – particularly in a resource-constrained mobile device – sensors themselves may vary in their availability and quality. Sensor range and quality are import properties to consider in the design of mobile context aware systems.

1.1.2 Interactive Intelligence

During our introduction of UbiComp, context awareness and their dependence upon work from the fields of AI and HCI, we briefly mentioned interactive intelligent systems (IIS), which are intelligent systems that users interact with [105]. Though IIS are certainly not restricted to mobile systems, the notion of an interactive intelligent mobile system allows us to elegantly state two important problem areas in mobile context aware systems: the intelligence, i.e. the AI component, which relates to context inference and learning processes; and the interaction, i.e. the HCI component, which relates to device interfaces and modes for interaction with the intelligence.

The Intelligence

Automatically inferring context with mobile devices is non-trivial, particularly if available sensors are unreliable or of variable quality. Further difficulties may be encountered if a mobile context aware system is expected to react to context changes in real time, which is probable given the need for *relevance* in context aware services and applications [57]. Sensor availability and resource limitations may impact on context inference further, e.g. attempting to use radio sensors such as GPS in areas of variable coverage, or using context inference approaches with large resource demands.

The aim of a context inference process is to achieve ‘good’ inference performance. Ideally, inference should be perfect, i.e. context is correctly classified at the correct time, but, in reality, this is rarely the case. The implications of incorrect inference may vary in their severity, e.g. a wrongly inferred location could be serious for navigation applications, but less serious for weather applications. We should therefore accept that context inference will occasionally be incorrect, and we should design the intelligence to *learn* about context over time in order to improve and maintain future inference performance.

The key questions then, are *how do we infer context?* and *how do we learn about context?* Can we enlist the help of the user in the learning process? Moreover, *how can we know when to infer context?* Real time context inference (and learning) is desirable, but identifying the correct time to infer is an inference problem in itself. It is also pertinent for mobile context aware systems, where resource constraints call for sensible inference and learning approaches.

The Interaction

Given intelligence that can infer and learn about context through a mobile device, how might we design for user interaction? As we proposed in the previous section, enlisting the help of the user in the context learning process may be prudent, but how should we elicit this help? Unlike desktop users – whose primary attention may be on a screen upon which prompts can be raised – mobile device users will not continuously interact with their device; nor are they likely to be diligent in helping an intelligent system if they feel it is asking for help too often. This raises an interesting question: *given that context learning could benefit from users' help, and given that users will only occasionally be able or willing to provide this help, how do we design mobile interfaces and interaction modes to best elicit this help?*

1.2 Research Contributions

Given the dissertation scope outlined in the previous section, our contributions in this dissertation are as follows:

- In relation to context sensing, we outline a set of sensors that could be used in a mobile context aware system. In a case study, we consider the everyday calendar as a 'virtual'[4] sensor of context data, and show it to be a poor reflection of reality due to the inherent 'noise' of reminders and events that do not actually occur, and *ad hoc* events that do occur but not appear in the calendar.

- In a further contribution to context sensing, we show – again, through our case study of the calendar as a context sensor – that performing low level data fusion of context data can improve sensing performance. We fuse our calendar data with other forms of context data (namely location and social network data), and show that this significantly improves context sensing performance.

- For the intelligence component of interactive intelligence in mobile context aware systems, we model context as discrete states in a finite state machine (FSM). This characterises two problems, the first of which is the inference of context transitions. Through a case study of place awareness [5], we design and analyse a system to infer place transitions in real time; showing how performance varies according to the system's parameters, and that good inference performance can be achieved using this simple approach.

- The second problem characterised by an FSM model of context is the inference of the context states. We contribute a context inference algorithm that is executed at the moment of context transition. This algorithm also measures the confidence of its reasoning and prompts the user for help – or *feedback* – if this measure is too low. It then learns from user feedback in real time using a branch of machine learning known as active learning [210]. Continuing our case study of place awareness, we apply the algorithm to the problem of real time place inference and learning. Using data collected during a field study, we simulate the algorithm's performance over a range of expected user feedback scenarios. The simulations are then used to compare a set of implementation designs for the algorithm; showing how a probabilistic inference approach gives superior performance to a geometric one.

- For the interaction component of interactive intelligence in mobile context aware systems, we develop a set of user feedback and interaction requirements. Continuing with the place awareness case study, we use the requirements to design a set of interfaces that request and enable user feedback for active learning. We deploy our whole system in a field study and show that places are inferred and learned well, even with the small amount of feedback provided by users. We also compare user response behaviour to different feedback request approaches and prompt modes; our results suggest that the use of speech prompts rather

[4] The concepts of 'physical' and 'virtual' sensors are outlined in Chapter 3

[5] The problem of place awareness is introduced in Chapter 4

than simple audio prompts does not significantly improve feedback response rate or time. Furthermore, the results suggest that requesting feedback actively, i.e. using audio, visual and tactile modes, rather than passively, i.e. using the visual mode only, does not significantly improve feedback behaviour for speech prompts but it does for simple audio (non-speech) prompts. Finally, we compare the field inference performance with the simulation performance from the previous study and show that the simulation is a reasonable approximation of user behaviour in the field.

These concrete contributions go some way to improving the design and implementation of mobile context aware systems. The next section outlines the dissertation structure.

1.3 Dissertation Outline

For the final section of this chapter, we outline the structure of the dissertation:

- Chapter 2 presents the background to context awareness and mobile context awareness. We review the relevant literature and active research areas before using them to derive a set of research questions that motivate the work in this dissertation. We also note the popular use of layer models in the context awareness literature, and present a layer model of our own to illustrate how our work fits together. This layer model will be used to guide the dissertation.

- Chapter 3 addresses context sensing in mobile context aware systems. This chapter is concerned with identifying sources and sensors of context data in a mobile environment, and we present a case study of the everyday calendar as a context sensor. During the study, we compare the calendar against actual events and show that – standalone – it is a poor context sensor. However, by fusing it with other sources of context data, the useful data can be extracted and performance can be increased significantly.

- Chapter 4 moves into interactive intelligence in mobile context aware systems. Focusing on the intelligence component, this chapter approaches the challenges surrounding context inference and learning in mobile context aware systems. Using a FSM model of context, we first address the problem of inferring significant context state transitions with mobile devices; using a case study of place awareness to show that device motion can be used to infer the majority of place changes in real time. We present our algorithms for context inference and active learning, and apply them to the place awareness case study.

- Chapter 5 focuses on the interaction component on interactive intelligence in mobile context aware systems. Here, we address challenges related to the elicitation of user feedback for active learning of context in the field with mobile devices. We deploy our mobile place awareness system in a field study to better assess its performance in the wild, and to compare alternative approaches to feedback elicitation. We report on our findings and conclude the chapter with a brief review of possible applications for our work.

- Chapter 6 concludes the dissertation by summarising our contributions and linking them back to our research questions and the overarching thesis question. We critically analyse how well the research questions have been addressed, and pay particular attention to the general implications and limitations of the work. Finally, we outline possible and alternative approaches for future work given the work presented in the dissertation.

Chapter 2

Background and Related Work

Context and context awareness are extremely broad research topics that span multiple disciplines within computer science. This chapter introduces, explores and reviews the literature surrounding the concept of context, context awareness and the narrower yet growing field of mobile context awareness. We also review the emerging field of interactive intelligent systems (IIS). The topics are introduced in descending order of granularity, and are designed to set the scene for the remainder of the dissertation.

At each stage of this literature review, we describe common research problems and issues, and attempt to summarise the state of the art for the areas that are relevant to our work. We use these to formally define our high level research questions, before introducing our layer model that will structure the technical chapters.

2.1 Context

We begin with the definition of context, and how it is interpreted by computer scientists. There are, in fact, multiple definitions of context in the literature, each developed according to the original applications that researchers had in mind. Here we list some of the more popular definitions:

- Schilit and Thiemer [204] first introduced the term 'context aware' within their work surrounding mobile distributed computing; specifically applying it to the problem of location awareness in an office environment. Their definition comprises of people's locations and identities, as well as the state of objects within their environment – which is further refined in their work on context aware computing applications [202] to include accessible devices and changes in people and devices *over time.*

- Ward *et al.* [235], while working on embedded sensor systems within the home, defined context to primarily be the location of an object in an environment.

- Pascoe *et al.* [174] first proposed that context was more than location – extending it to include environmental features such as, for example, the current weather description. The idea of context being 'more than location' was further outlined by Schmidt *et al.* [208], in which they propose context to include environmental conditions and infrastructure, as well as information about devices, users and user tasks.

- Chen and Kotz [38] further include the context of time, defining context to be "*the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user*". They further categorise context into two categories: *active* context, which influences application behaviour, and *passive* context, which is peripheral yet still relevant to the application.

- Lieberman and Selker [138] have defined context as "*everything **but** the explicit input and output*" of an application, specifically the state of the user, as well the states of the physical and computational (virtual) environments. They also include the history of interactions between each.

- Dey and Abowd [57, 59] define context as "*any information that can be used to [characterise] the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*".

- More recently, Zimmermann *et al.* [246] have attempted to extend Dey's definition into categories of entity information, namely: location, time, individuality, relations and activity. They also conjecture that context should be defined by its *use*, i.e. its interpretation, and the transitions between contexts over time.

General definitions of context are vague and application specific, but the definition by Dey [57] is seen as the *de facto* standard within the ubiquitous computing community. It is both abstract in its description, e.g. "*context is **any** information that [characterises] the situation of an **entity*** [...]", and specific in its domain, e.g. "[...] *considered relevant to the interaction between a user and an application* [...]". Due to these properties, and its popular adoption in the literature, we will follow this definition of context throughout the dissertation.

Following Dey's definition, there is a myriad of information that can be used to describe a user's situation, e.g. where they are (their location), what they're doing (the activity), whom they're doing it with, when they're doing it, what they're using to do it, where they're intending to go next, where they've been, etc. Each of these somewhat abstract categories can be further described using concrete examples, e.g. location may be described by degrees latitude and longitude or by user-defined interpretations (e.g. "home" or "work"); activity may be described generally (e.g. "travelling") or specifically, as relevant to the user (e.g. "walking to work").

As these examples illustrate, users can interpret their context differently from others, who in turn may describe their own context in a different manner. An apparently objective description of the same context may differ again, e.g. the user describes her location as "the office"; her friend, referring to the same location, may describe it as "in London"; and a GPS sensor might describe it as 51.5049672, -0.0197931. Part of the reason that context is so vaguely defined and application-specific [246] is the subjective nature in which it can be defined in practice [62].

The work in this dissertation is chiefly influenced by Dey's definition of context, as it avoids the application specific nature of other work, e.g. [235]. We note, however, that even Dey's definition is hard to develop from the abstract to the concrete without a classification or modelling system. Dey and Abowd extend Dey's definition into a model, as we shall see in the next section.

2.1.1 Classifying Context

Once we have defined our interpretation context, how should it be categorised or classified? That is, how can we formally and systematically label context such that it can be useful in computing applications? This question has led to a range of context classification systems, ontologies and models within the literature, and – as with context definitions – these are typically driven by technology and applications. Again, there is no agreed standard classification system, so here we summarise some of the key approaches.

Dey and Abowd, following their definition of context, argue that the key categories of context are location, identity, time and activity [58], which extends the physical/user environmental distinction proposed by Schilit *et al.* [202]. Schmidt *et al.* [208] support the environmental distinction, but extend the granularity of the physical environment. More recently, Zimmermann *et al.* [246] have extended Dey and Abowd's approach to include a social aspect, i.e. incorporating entity relationships (see Figure 2.1), and a further temporal aspect: context transitions, i.e. the idea that context classification changes over time. Following the popularity of motion sensing in context aware systems (cf. Section 2.4.4), Chalmers [35] extends the Dey and Abowd classification to include motion and environment.

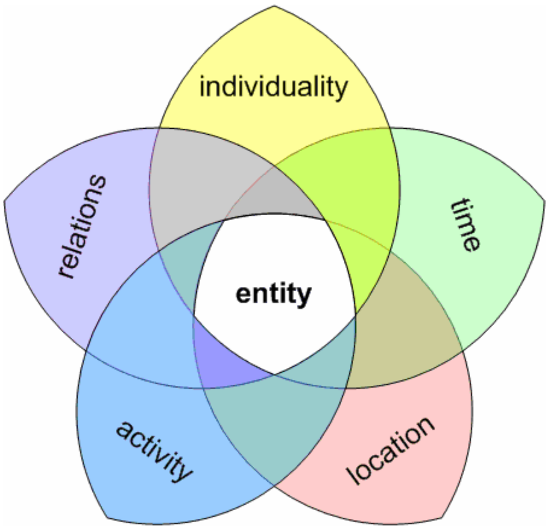


Figure 2.1: *Zimmermann's context categories surrounding the Dey and Abowd categorisation [57] (Image source: [246].)*

Although there are multiple and fragmented approaches to classifying context in the literature, there is common agreement on certain aspects of context. The two most common categories are location and activity. Almost all attempts at classification include them, and they have been developed into research sub-fields in their own right (location awareness and activity recognition).

2.1.2 Context Facets

Here we link definitions and categorisations of context to the Five Ws approach presented in the introductory chapter of this dissertation. As we shall see, researchers have used this approach in the past due to the natural description of context that the Five Ws provide.

The Five Ws Approach

To remind the reader: in journalism, a "Five Ws" interrogatory approach is often used to gather information about an event. Referring to who, what, where, when and why questions, the Five Ws mnemonic is both intuitive and useful when applied to many information gathering scenarios. It is natural therefore, to ask whether the Five Ws model can be used to describe and categorise context.

The Five Ws – sometimes referred to as 5W1H – has indeed been used for this purpose. Abowd and Mynatt recommend using the approach as the very first step in context aware system design [2] – highlighting the lack of standardisation in the definition and

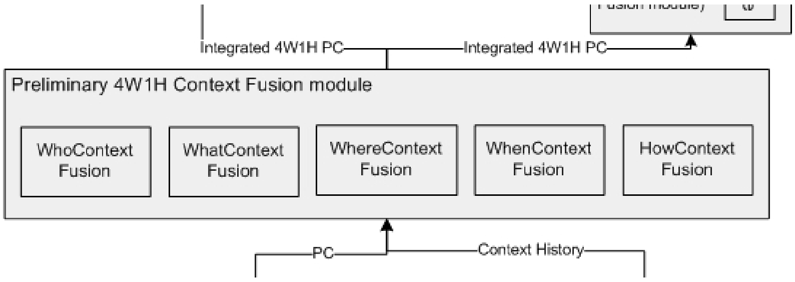


Figure 2.2: *Oh, Yoon and Woo's 4W1H interpretation of context facets, from [166].*

classification of context. Dix *et al.* apply the approach to location and space questions in their design framework for interactive mobile systems [61]. Further design models using variants of the Five Ws have been used by Oh *et al.* for mobile devices [166] (see Figure 2.2) and ubiquitous computing in smart homes [239].

Following Abowd and Mynatt, we use the Five Ws approach to define a set of context *facets*:

- **Who**: e.g. "Whom are we trying to identify?"'; "Who is using our application?"; "Who generated this context data?". The "Who" facet refers to user identity, one of the key categories of context proposed by Dey and Abowd [59]. Although typically used in an individualistic manner – i.e. the identity of a single user, namely the user in Dey's definition – the "Who" facet can also extend to other people that may be relevant to the user's situation, e.g. friends in a social network, co-located people, or other users of an application.
- **What**: e.g. "What is the user doing?". The "What" facet refers to activity, another key category of the Dey and Abowd approach. Activity is fundamental to many context models, e.g. [57, 58, 246] and activity recognition is a popular and fast-moving research field within the UbiComp community [16].
- **Where**: e.g. "Where is the user?"; "Where is the device or object?". Location is by far the most popular category of context, due in part to its use in commercial applications in recent years, e.g. map applications on mobile devices. As we saw in the previous section, early definitions of context focussed primarily, and in some cases almost entirely, on location [235]. This focus changed, however, as researchers began to realise the value of other context categories beyond location [208].
- **When**: e.g. "When is the user doing this?"; "How long will the user be doing this for?"; "How long will they be there for?". The temporal aspect of context has been recently explored in the research of routine and patterns in people's daily lives [67]. Temporal context is often related to changes in other context categories

over time, e.g. [19, 246], and can add complexity to the context awareness, e.g. how can we capture and model context changes over time?

- **Why**: e.g. "Why is the user doing this?"; "Why is the user here?". Much like the "When" facet, the "Why" facet is typically related to other categories, e.g. "Why this activity?" or "Why this location?". This is perhaps the most complex context facet to analyse, as we would have to consider, for example, action meaning, intent or emotion [239, 166]. Emotional context in particular is non-trivial to interpret [187].

Although the Five Ws were developed in the context of journalism, we feel that their breadth captures much of what context is about. Dey and Abowd's work using location, time, identify and activity is certainly broad, but the Five Ws are more abstract and we see the Dey and Abowd facets as instantiations of each, e.g. identity is an instantiation of 'Who'. Much of the work in this dissertation is arguably better classified using the more abstract definition than the concrete one, and hence we make use of the Five Ws model throughout.

The Five Ws as a Theory for Context?

Our earlier discussion of context as its definition highlighted the fact that there is no standard definition of context, even though many varied definitions exist. As such, there is no standard theory of context either, and researchers develop definitions and theories around their applications. Dey's definition [57] is a *de facto* definition – used and cited often – but researchers do still produce new definitions, e.g. [35].

This raises the question of context theory, and how the UbiComp community might work towards developing a standard theory for practitioners to adopt. We conjecture that the Five Ws, although a journalistic heuristic, is a sound basis upon which to build such a theory, and the work in this dissertation aims to advance this notion. In the concluding chapter, we will discuss how the work has advanced the idea of a context theory and possible paths for further development.

2.2 Context Awareness

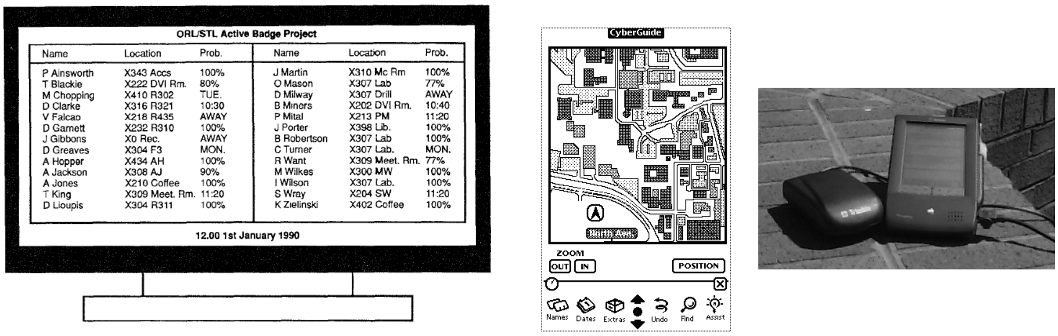
Given the philosophical and theoretical notions of context, how might we tangibly exploit it – as computer scientists – in the systems that we design? In addition to how, we should also ask *why* knowledge of context might be useful to our system and its users, and why we should go to the bother of obtaining it. The main benefit that context knowledge provides is *relevance*. Computer systems should be designed with a benefit in mind and, particularly in HCI, that benefit should be to the user of the system. If a computer had knowledge of its user's context, then it could enable applications and services that are relevant to the user at any given time. This could potentially make people's lives easier by reducing burden associated with tasks such as search or navigation. A computer system that can obtain and utilise knowledge of context in this manner is defined to be context aware [202].

The importance of context awareness – or context aware computing – in computer science has increased in recent decades as computers have become ever more pervasive in everyday life. As we saw in the previous section, context has no strict definition, and its interpretation can vary depending on application. The idea of computers sensing and reacting to a user's situation has been a popular research topic for a number of years, featuring regularly in computer science conferences and journals and occasionally in commercial products. The vision of pervasive computing integrating into the environment – functioning only when necessary and without obstructing or annoying the end user – is certainly not a reality yet, but technology is moving incrementally closer to a world of "computing everywhere" [236].

As we saw in the introductory chapter, context awareness draws upon areas from many other fields in computer science and engineering. For example, some of the key ideas behind artificial intelligence – agents sensing and reacting to environments, knowledge representation, inference, reasoning, learning and planning – interweave with the desirable traits of a context aware system [21]. Relations to other fields include: HCI (ubiquitous computing, user interfaces and user-centred design) [62, 205]; telecommunications (sensors and wireless sensor networks); and mathematics (statistics, inference, data structures and algorithm design).

2.2.1 Notable Context Aware Systems

One of the earliest and perhaps most recognised context aware system is the Active Badge location system [234] (Figure 2.3a). Using a wireless sensor network deployed within an office environment, workers' locations are sensed through body-worn RF badges. Designed to assist the redirection of phone calls to the relevant employee's nearest desk phone, the system provides basic location awareness with a probabilistic measure of confidence. The active badge system has been pioneering, and its basic approach – deploying sensors and sensor networks to enable context awareness – has been adopted by many researchers since, e.g. [1, 14, 58, 67, 145, 168, 204, 235]. Early propositions for context aware systems included a location aware shopping assistant that used RF beacons to guide and assist customers around shops [12].



ORL/STL Active Badge Project

Name	Location	Prob.	Name	Location	Prob.
P Ainsworth	X343 Accs	100%	J Martin	X310 Mc Rm	100%
T Blackie	X222 DVI Rm.	80%	O Mason	X307 Lab	77%
M Chopping	X410 R302	TUE.	D Milway	X307 Drill	AWAY
D Clarke	X316 R321	10:30	B Miners	X202 DVI Rm.	10:40
V Falcao	X218 R435	AWAY	P Mital	X213 PM	11:20
D Garnett	X232 R310	100%	J Porter	X398 Lib.	100%
J Gibbons	X0 Rec.	AWAY	B Robertson	X307 Lab	100%
D Greaves	X304 F3	MON.	C Turner	X307 Lab.	MON.
A Hopper	X434 AH	100%	R Want	X309 Meet. Rm.	77%
A Jackson	X308 AJ	90%	M Wilkes	X300 MW	100%
A Jones	X210 Coffee	100%	I Wilson	X307 Lab.	100%
T King	X309 Meet. Rm.	11:20	S Wray	X204 SW	11:20
D Lioupis	X304 R311	100%	K Zielinski	X402 Coffee	100%

12.00 1st January 1990

(a) *The ActiveBadge system was one of the first operational context aware systems. (Image from [234]).*

(b) *Abowd* et al.*'s Cyberguide: a mobile context aware tour guide. (Image from [1]).*

Figure 2.3: *Notable early context aware systems.*

In 1997, Abowd *et al.* introduced their Cyberguide system [1] (Figure 2.3b), a context aware tour guide that used real time location and location history to guide people around, for example, a museum or tourist attraction. (Museum and exhibition tour guides have since become popular applications for context aware systems, e.g. [40, 221, 247].) The Cyberguide project was adapted into Dey and Abowd's CybreMinder system [58], which used their Context Toolkit [55] to deliver context-triggered reminders to people, e.g. items on a to do list. The Context Toolkit was also used with various prototype applications, including an input/output (IO) board – which was a primitive presence and availability system designed to operate in workplaces – and a conference assistant, which guided the user around a multi-track academic conference based on their location and academic interests.

Other early context aware systems include: "Everywhere Messaging", a message system that attempts to deliver messages between users through multiple modalities and mediums based on user context; ConChat [189], a context aware chat program that was designed to replicate face-to-face conversation using context data sensed from the conversation participants; SmartRestaurant [147], a system that used customer context to improve the efficiency of food ordering in restaurant; and Coordinate [97], a system that used calendar data to predict workers' likely meeting availability. Context data has been utilised by researchers beyond those exploring standalone systems. Begole *et al.* have explored visualisation of people's temporal patterns in collaborative computing using context data [19], Muñoz *et al.* [158] have shown how context aware computing can aid collaboration in a hospital environment and Hudson *et al.* [101] used context data to analyse people's reactions to interruption on desktop systems.

Context aware computing research has also been conducted within so called "smart home" environments. Smart homes are domestic environments that have been augmented with sensors and computing devices, and the data collected has been used for occupant modelling, e.g. [193, 223, 229], multimedia delivery, e.g. [99] and – more recently – energy monitoring and usage, e.g. [53].

The systems presented so far are, in the majority, desktop-based, i.e. there is little or no use of mobile devices, and many require bespoke hardware to function. In recent years, the focus in context awareness research has shifted to the use of mobile devices such as smartphones or tablet PCs; due in part to the enormous popularity and ubiquity of such devices in people's daily lives. This shift has resulted in a sub-field of context awareness: mobile context awareness.

Desktop systems *are* still important however, and research conducted on or using them has influenced the work in this dissertation. In particular, Horvitz *et al.*'s work using desktop calendars [97] has informed our work on using the calendar as a sensor, and integrating desktop systems with mobile ones.

2.3 Mobile Context Awareness

Mobile context aware systems are context aware systems in which mobile devices play a significant role in enabling context awareness.

One of the most ubiquitous tools in the progress of context awareness has been the mobile device. Its enormous popularity and permeation into daily life – coupled with increasingly sophisticated hardware – has greatly increased the potential for context awareness outside research environments. The very mobility of these devices is key to the idea of mobile context awareness, where the sensing and inference is enabled by – and even conducted upon – the device itself. It is both a sensing platform and a computer, and the relentless increase in mobile computing power and sensing capability – motion sensors, light sensors and multiple radio sensors come as standard in the modern smartyphone – allow for a whole new area of mobile context awareness research and development.

Today, mobile device users are becoming used to "always on" network connectivity; taking advantage of faster connections to use services such as push email, synchronised calendars and online application programmable interfaces (APIs) into social media services, e.g. Facebook. These 'virtual' sensors can expose the mobile device to additional data sources such as social networks, user preferences, tagged photographs and music playlists. Fusion of these sources with traditional 'physical' sensors, e.g. GPS, can allow for better inference of context and, subsequently, a wider range of mobile applications that utilise context. Furthermore, software developers have been turning to mobile devices for their application development. The soaring popularity of services such as Apple's App Store and the Google Play Store means the mobile application business is predicted to be worth $17.5 billion in 2012 [211].

Many of the currently available context aware mobile services are limited to being "location-based services"; they focus primarily on the device's location, the user's interaction with the device and the services that the location-aware capability can enable, e.g. navigation. As we discussed in Section 2.1.2, location is a key feature of context, but it is not the only one; and this is even more apparent with the increasing range and diversity of data available to the typical mobile device.

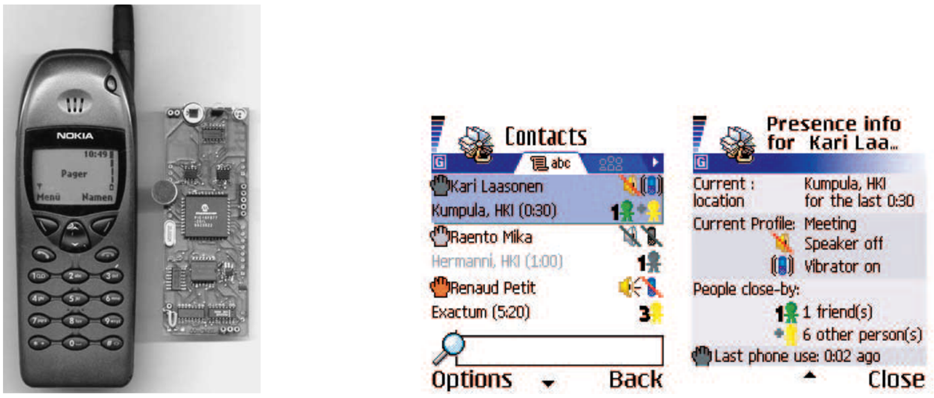
The potential for mobile context awareness is encouraging – the mobility of the device can allow for the sensing and reaction to users' everyday situations with little or no specialist hardware and relatively simple system architectures. This mobility comes at a price however: mobile devices are typically resource constrained – battery power, CPU limitations and network connectivity must be traded off against the demand for accurate and usable context aware awareness. These trade-offs, coupled with the potential applications of – and improvements to – mobile context aware computing methods, offer many challenges to the research community.

In this section, we review key literature within the field of mobile context awareness; highlighting notable early systems. It is here that we build up to the state of the art in mobile context aware computing, which will be discussed in the subsequent section on active research areas.

2.3.1 Early Mobile Context Aware Systems and Applications

Following the evolution of desktop-based computing to ubiquitous computing, context aware systems have made use of mobile devices. Early context aware systems such as Abowd *et al.*'s Cyberguide [1] and Schmidt *et al.*'s work with primitive PDAs [208] used mobile devices primarily for the location awareness. Chen and Kotz, in their survey of early mobile context aware systems [38], reviewed a set of projects that used mobility in context awareness. The survey showed that – although mobile device use in context aware computing was gaining – mobility was simply a provider for basic location and time data. Although still well cited today, this survey is perhaps a little outdated given the surge of smartphones and tablets in recent years.

Dey *et al.* began to incorporate mobility into their work with the Context Toolkit [55], and Schilit *et al.* highlighted the important role of mobile devices in context aware communication [203]. Gellerson *et al.* began to explore the utility of mobility when deploying mobile sensors into users' natural environments and artefacts, e.g. a coffee mug [78]. Hofer and Schwinger [93], and Chen *et al.* [39] presented new architectures for context aware computing, both of which made extensive use of mobile devices, and



Contacts
abc
Kari Laasonen
Kumpula, HKI (0:30) 1
Raento Mika
Hermanni, HKI (1:00) 1
Renaud Petit
Exactum (5:20) 3
Options Back

Presence info for Kari Laa...
Current : location Kumpula, HKI for the last 0:30
Current Profile: Meeting
Speaker off
Vibrator on
People close-by:
1 1 friend(s)
6 other person(s)
Last phone use: 0:02 ago
Close

(a) *The TEA project integrated additional sensors into early, commercially available mobile devices. (Image from [78]).*

(b) *ContextPhone was the first mobile context awareness prototyping platform. (Image from [188]).*

Figure 2.4: *Notable, early mobile context aware systems.*

Henricksen *et al.* [87] presented an abstract approach to modelling context information in pervasive (particularly mobile) context aware systems.

Siewiorek *et al.* introduced their context aware mobile phone – SenSay [212] – which was an early standalone context aware mobile device developed prior to the smartphone era. Korpipää *et al.* – recognising the emerging need for formal management in the development of mobile context aware systems – designed a framework for mobile context management that uses a formal ontology of context sensor and source types [122, 123]. The first mobile context prototyping platform, ContextPhone, was released soon after by Raento *et al.* [188]. ContextPhone (see Figure 2.4b) identified a set of common developmental areas for mobile context aware systems, namely: sensing, communication and application. ContextPhone was a very influential project, used by later large scale mobile context aware systems such as Reality Mining [67]. The ideas behind ContextPhone – particularly the layered architecture and its ability to both sense and infer on-device – have provided a foundation for much of the work in this dissertation.

The influx of data that mobile devices could supply to context aware computing applications – and the potential noise and ambiguity that could arise from large quantities of data – led to research into context *mediation*, whereby available context data is intelligently selected given an input request, usually from the user. Chalmers *et al.* [36, 37] performed extensive research into formal context mediation in mobile devices, using a case study of map zooming to illustrate the benefits of mediation. Dey and Mankoff [60] subsequently presented a design framework for context mediation in mobile devices.

Meanwhile, researchers were beginning to employ wireless infrastructure into mobile context aware systems. Krumm and Horvitz [126] used WiFi signal strength fluctuations to infer user location and movement patterns, and Laasonen *et al.* [129] presented an on-device location recognition framework that was based, interestingly, around user-defined locations (rather than machine-inferred locations). Although somewhat limited by the cellular infrastructure available at the time, Laasonen *et al.*'s work showed remarkable generality, which is reflected in similar, later work on place recognition, e.g. [90, 117]. Indeed, Laasonen's unsupervised learning approach has served as inspiration for the work on place inference and learning in this dissertation. The ActiveCampus mobile context aware system is a notable example of mobile context awareness in the field. Developed by Griswold *et al.* [81], its ActiveMap component was one of the first mobile context aware systems to incorporate user feedback into the context inference process.

Researchers then began to explore larger datasets sourced from mobile devices. One of the most influential of these is from Eagle and Pentland's Reality Mining study [65, 67], which collected and analysed Bluetooth and other RF data from 100 subjects over the course of a year. This publicly available dataset not only provided insight into people's daily lives, e.g. temporal patterns associated with routine, but also their social and application behaviours. This dataset has been further analysed by researchers, leading to findings associated with, for example, principal behaviours – or 'eigenbehaviours' [66] –, recommender systems [108], context prediction [213] and activity prediction [47].

Reality Mining is perhaps the canonical example of the richness of data that can be obtained through mobile context awareness. Its approach – Bluetooth sensing – is similar to the one adopted by us for our calendar study, and the fact that it was performed using older mobile technology, its results and its datasets are still being used for research today.

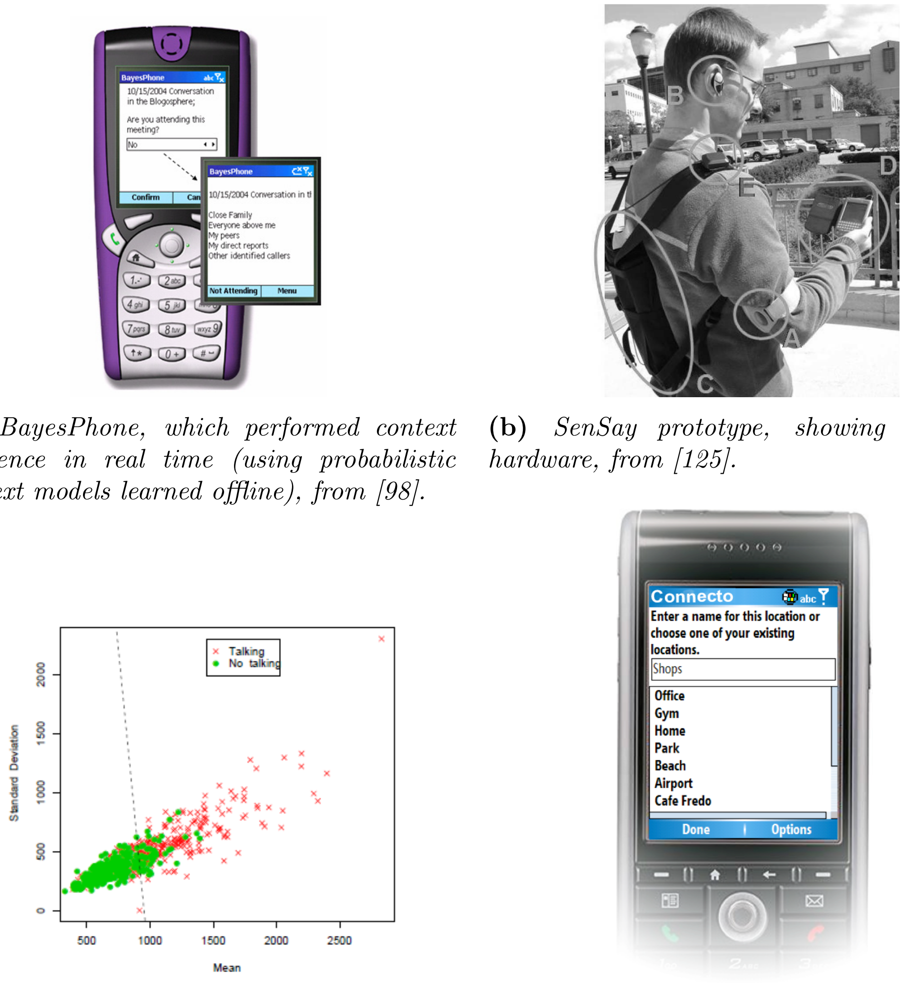
The Reality Mining study showed how rich context data could be obtained using commercially available mobile devices (rather than bespoke hardware). Similar large scale studies with mobile devices followed, including: PlaceLab [130], a large scale approach to location positioning using existing RF infrastructure in the wild; and Cityware [168], which studied behaviour at the city scale using RF data obtained using mobile devices.

As mobile devices became more ubiquitous and powerful, researchers began to study the feasibility of performing context inference on-device using learned models. The BayesPhone, developed by Horvitz *et al.* [98], used pre-trained models of user behaviour, e.g. call handling, that operated on-device. Although BayesPhone required offline training, its key advantage lay in user customisation. By incorporating users' own schedules and behaviour patterns into the machine learning process, it could improve the *relevance* of its services to the user. Hightower *et al.* [90] explored on-device context learning in their BeaconPrint project exploring place awareness. Whereas BayesPhone used mobile devices for context inference with offline learning, BeaconPrint performed learning and inference online with the disadvantage of requiring additional hardware, i.e. a laptop, to operate. Krause *et al.* [125] – building on earlier work from the SenSay system [212] – used a wearable sensor array that integrated with a mobile context aware device to perform online learning of users' personal preferences. As with BeaconPrint, this system required additional hardware to perform context inference and learning in real time.

CenceMe [154] was the first mobile context aware system to infer, learn and share context online through users' social networks. By combining on-device context classifiers with more heavyweight offline classifiers, CenceMe was able to perform near real time inference with a favourable user experience using commercially available mobile devices. CenceMe was well designed with a very large sample set, but it somewhat compromised on its execution by offloading much of the hard inference work offline. Although this was done for practicality reasons, we feel an interesting question relates to the capability (and possibility) of tasking the device with most (if not all) of the inference and learning. How might it perform, would it reduce awareness latency and what are the implications for device resources?

Researchers were also studying the effects of integrating mobile context aware devices into people's everyday tasks and routines. The Place-Its system [214] used mobile devices to deliver reminders to participants when entering and leaving important locations, and Comedia [104] used mobile context aware devices for media capture during social events. A particularly interesting study was the Connecto system by Barkhuus *et al.* [17]. Connecto used participants' mobile devices to both capture and share location within social groups. The findings were interesting due to their far reaching implications; insight was gained into: how people label locations (naming by *place* rather than space, i.e. geographic location, or activity was the most common practice); how people co-ordinate and communicate within groups; and how information sharing evolves over time in a 'storytelling' style. Connecto has been a big influence on the later work in this dissertation. It is one of the better examples of 'place' vs 'space' in practice, and has been a useful source evidence for people's reasoning behind the *meaning* of their places.

As for other mobile context aware systems, Froehlich *et al.* [73] presented MyExperience, a mobile context aware system that captured objective sensor data and subjective user experience feedback through mobile devices. Designed to aid the capture of data by researchers in the field, MyExperience used active prompting to elicit feedback from users on their current experience, whilst concurrently logging passive sensor and application data through the device. Similar projects included: UbiFit Garden [50], a mobile context aware system that used inferred user activity and a virtual 'garden' to encourage people to increase their physical activity; and UbiGreen [74], a mobile context aware system for encouraging more environmentally friendly transportation



10/15/2004 Conversation in the Blogosphere;

Are you attending this meeting?

No

10/15/2004 Conversation in tl

Close Family
Everyone above me
My peers
My direct reports
Other identified callers

Enter a name for this location or choose one of your existing locations.

Shops

Office
Gym
Home
Park
Beach
Airport
Cafe Fredo

(a) *BayesPhone, which performed context inference in real time (using probabilistic context models learned offline), from [98].*

(b) *SenSay prototype, showing wearable hardware, from [125].*

(c) *Real time, on-device inference (voice classification) in CenceMe; from [154].*

(d) *Labelling meaningful places in Conneto, from [17].*

Figure 2.5: *Notable mobile context aware systems.*

habits.

The next section focuses on modern research in mobile context aware systems, most of which is influenced by the work reviewed in this section. By dividing the research into key areas, we will review important and influential work that leads to the state of the art in mobile context awareness research.

2.4 Relevant Active Research Areas

In this section, we review the active research areas within mobile context awareness. Much of the work in the previous section has led to a set of research trends in the field, and we categorise the work according to these trends.

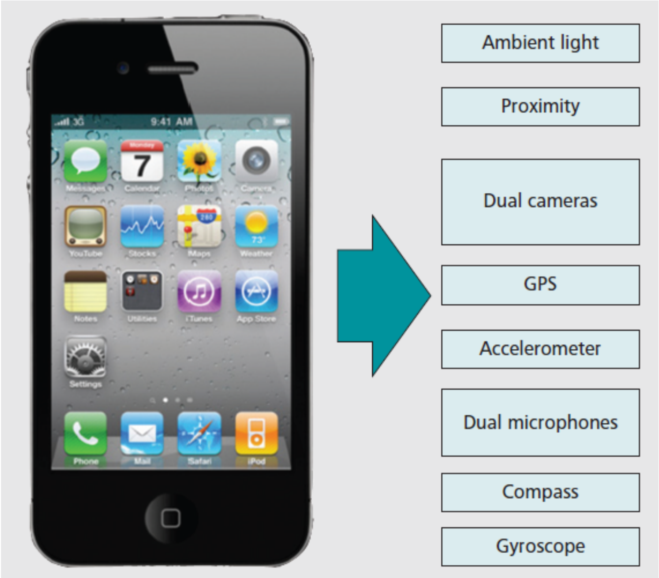


Figure 2.6: *Sensor list from a modern iPhone 4, (Image from [131]).*

2.4.1 Mobile Context Sensing

The problem of acquiring context data using mobile devices is an ongoing research topic within mobile context awareness, particularly in relation to resource efficiency and context inference accuracy [23]. Although mobile device capabilities have evolved rapidly in recent years, there are still many sources of context data that are not being exploited by mobile context aware systems.

In a recent survey of mobile phone sensing, Lane *et al.* [131] list a set of typical hardware sensors that are found on an off-the-shelf iPhone 4:

- Ambient light
- Proximity
- Dual cameras
- GPS
- Accelerometer
- Dual Microphones
- Compass
- Gyroscope

In addition to these, mobile devices contain multiple RF technologies such as WiFi and Bluetooth as standard, and further environmental sensors such as temperature, pressure and humidity sensors have been developed for Android-based handsets [1]. Although modern smartphones are ubiquitous, and although researchers can take advantage of this ubiquity through scalable and rapid deployment of prototype systems, the key issue in mobile context sensing is the fact that – aside from GPS – sensors are typically *implemented for purposes other than context sensing*. For example, accelerometers are primarily used for screen orientation; ambient light sensors for screen illumination; cameras for image capture; microphones for communication; WiFi for network connectivity; and Bluetooth for media sharing. Researchers and developers must therefore attempt to exploit this infrastructure for their own applications.

Given these restrictions, interesting and useful work has been undertaken with regard to sourcing and sensing context data in the wild. For location awareness, Indulska and Sutton [103] first introduced the distinction between physical, virtual and logical location sensors. This distinction is not only a useful one for modelling traditional sensors; it allows us as researchers to model a large range of data sources as 'sensors', even though they are actually sensors in the traditional sense. Baldauf *et al.* [15] expanded the generality of this distinction and argued that almost anything in a user's physical or virtual environment – providing it can transduce a data source into machine readable data – could be considered as a potential sensor of context. At present, research has been undertaken into the use of biological entities [127], calendars [145], images [41], ambient sound [13, 146], screenplays [45] and social networks [18] as context sensors.

These sometimes unusual approaches to sensing have yielded interesting results which are relevant to the work in this dissertation. Given the enormous amount of data available to a modern mobile device through applications and websites, the exploration of virtual context sensing should continue. We aim to further this avenue of research in our work on using the calendar as a virtual context sensor.

Other approaches to sensing context with mobile devices have lead researchers to create hybrid sensing platforms that combine commercially available devices with custom designed sensor arrays. Similar to earlier work by Hightower *et al.* [90] and Krause *et al.* [125], the SeeMon system [111] attempts to integrate external sensors with mobile devices in order to provide rich datasets for analysis. Similarly, the embedded sensing platform (ESP), developed by Choudhury *et al.* [43], provides a set of external sensors that can integrate with mobile context aware systems, e.g. UbiFit [50] and UbiGreen [74].

[1] `http://developer.android.com/guide/topics/sensors/sensors_overview.html` (Accessed 2012-11-16)

In their review of mobile sensing [131], Lane *et al.* go on to highlight the growing demand for sensing context data on human behaviour and translating raw data into useful *meaning* through personalised sensing. The problem of sensing context data with mobile devices is further exacerbated by user behavioural diversity – which is studied at great length by Falaki *et al.* [68], who note the particularly large between-user ranges of interaction frequencies, duration and application types – and the variable proximity of users to their personal mobile devices at any given time [56, 175].

In summary, and using Indulska and Sutton's useful distinction, it is virtual context sensing, rather than physical sensing, that has the most potential for new research. As such, we pursue this avenue in Chapter 3.

2.4.2 Communication Sensing and Analysis

Following the somewhat traditional approaches to context sensing in the previous section, there is a large body of work that uses communications media to sense and analyse deeper context such as human emotion or intent. An example of this is sentiment analysis, in which computers attempt to infer human opinions from communications media such as Twitter, Facebook and email messages [172].

Pang *et al.*'s work used minimum cut sets in graphs to model film reviews and extract sentiment – in this case, a positive or negative review of a film – to good effect [171]. The authors use a very technical approach to an abstract problem, and the work is extremely thorough and effective. Its publication was an important step towards directly sensing and inferring sentiment from raw data.

Other applications of sentiment analysis have focused on voter opinions in political elections [165], the detection of aggressive content in emails [216], inappropriate content detection in online advertisements [107] and applications for recommender systems [224].

With the onset of large social media websites such as Twitter and Facebook, researchers have applied sentiment analysis techniques with varying effects. Agarwal *et al.* have recently attempted sentiment analysis with Twitter data with reasonable – but not exceptional – accuracy ($\approx 60\%$) [6]. O'Connor *et al.* have attempted to link sentiments in Twitter data to presidential elections with promising, but still only reasonable, accuracy results [165]. This works serves to illustrate how non-trivial sentiment and opinion analysis can be.

As for sensing emotions through mobile devices, the most active recent work is that of Rachuri *et al.* at Cambridge University and their work on the EmotionSense platform [186]. Here, emotions such as 'Happy', 'Sad' and 'Angry' are sensed and classified through mobile devices. EmotionSense can correctly classify these emotions ≈ 50–60% of the time – which is reasonable – but it suffers from noise, e.g. ambient sonic noise, latency issues and variability between subjects. The authors propose combining the system with bio-sensors such as galvanic skin response sensors in order to improve accuracy, but they acknowledge that key difficulties lie in the complexity of the problem rather than the implementation of the solution.

This is one of the reasons that this dissertation does not concentrate as heavily on the 'Why' context facet. Sentiment, opinion and emotion sensing is not easy, and yields variable results even with well designed systems such as EmotionSense. It is certainly and interesting and worthy research problem, but its difficult makes empirical research challenging and perhaps distracting from the themes in this dissertation.

Nevertheless, we feel that this area of research is one of the most fruitful for future work. Although EmotionSense reported some negative results, it has moved the idea of emotion and intent sensing a step closer towards realisation.

2.4.3 Location Positioning with Mobile Devices

Since early context aware systems focused primarily on location, e.g. [235], and the growing popularity of GPS-enabled commercial devices, location positioning has been an important issue in the field of mobile context awareness. One of the largest problems for researchers has been indoor location positioning. Because GPS does not work well indoors, various alternative approaches to the indoor location problem have been attempted, mainly based around the ActiveBadge model [234]. Example systems following this approach include: RADAR [14] which uses base station signal strength profiles; Cricket [182], which uses ultrasound; and LANDMARC [163], which uses RFID devices carried by users.

Because of its accuracy, we adopt a similar approach to ActiveBadge for our sensing study in Chapter 3. Much like Reality Mining [67], our 'badges' are Bluetooth mobile devices.

The main problem with the ActiveBadge approach, however, is the requirement for bespoke hardware, i.e. the deployment of custom sensors in the users' environments. There is something of a trade off between the work required for implementation and the reliability of the fine-grained location data obtained. There are also issues with scalability. Because of this, researchers have exploited existing infrastructure, e.g. assisted GPS (A-GPS), cellular networks and 802.11 WiFi access points, for indoor location positioning. Rekimoto *et al.*'s LifeTag system [192] uses each of these technologies in a city-wide (Tokyo) deployment of a location aware system, Krumm and Horvitz

[126] used WiFi signal strength to infer motion and position, Jiang *et al.* [106] use WiFi to fingerprint individual rooms within buildings and Kjærgaard *et al.* [120] use WiFi-enabled mobile devices to infer the indoor movement of pedestrian flocks. For a recent comprehensive review of location positioning technology in mobile devices, see [244]. Other approaches to indoor localisation have used ambient sound to 'fingerprint' locations [13], user motion and inertia [135, 240], relative positioning [77, 181] and geo-magnetic disturbances [48].

There is also an extensive range of commercial location positioning systems available. Google's myriad of location-based technologies, e.g. Google Maps, Street View (and emerging indoor Street View[2]), Android's location providers (which rely on anonymously collected data from Android mobile device users[3]) and Google Now (see Figure 2.7c) are perhaps the most familiar and popular. However, at the time of writing, a market for indoor location technology is beginning to emerge[4], with key companies such as Nokia, Microsoft, Apple, Qualcomm, Cisco and Motorola all developing indoor location positioning technologies for commercial use.

The popularity and demand for indoor location positioning has also generated a slew of startup companies. The two most popular are perhaps Shopkick[5], which uses ultrasound positioning technology similar to the Cricket system [182], and SkyHook[6], which fuses GPS, A-GPS and WiFi sensors for its location positioning.

In the academic space, however, the state of the art is *place awareness*, which moves beyond simple 'space' [86] into the inference of location *meaning* to individual or groups of users. Following Laasonen *et al.* [129], and stemming directly from Hightower *et al.*'s BeaconPoint mobile place learning and recognition system [90], Kim *et al.* have developed a place recognition approach that uses wireless fingerprinting [116, 117] to automatically recognise people's places using their mobile devices. Their current approach – SensLoc [117] (Figure 2.7b) – remains the state of the art for automated place capture and recognition with mobile devices, but the key problem researchers are facing is the elicitation of meaning and the incorporation of user feedback into the place recognition process [115]. This, coupled with the subjective nature of a 'place' [89] and the diversity of people's mobile interaction behaviour [68], sets a boundary for current research in location based mobile computing. The key problem with these systems, however, is the capture of place meaning from users. Both BeaconPrint and SensLoc rely on unprompted user input in order to label places. Without user input, places are

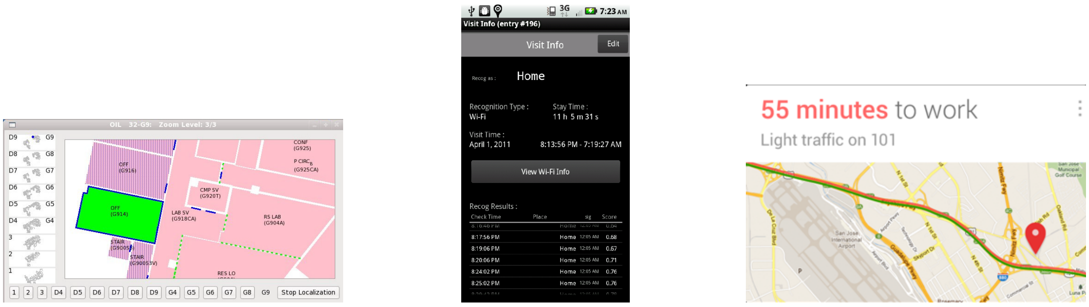
[2] `http://maps.google.com/help/maps/helloworld/index.html` (Accessed 2012-11-16)

[3] `http://developer.android.com/guide/topics/location/strategies.html` (Accessed 2012-11-16)

[4] `http://finance.yahoo.com/news/google-tops-abi-research-indoor-152600971.html` (Accessed 2012-10-03)

[5] `http://www.shopkick.com` (Accessed 2012-11-16)

[6] `http://www.skyhookwireless.com` (Accessed 2012-11-16)


Visit Info (entry #196)
Visit Info
Edit
Recog as : Home
Recognition Type : Wi-Fi
Stay Time : 11 h 5 m 31 s
Visit Time : April 1, 2011 8:13:56 PM - 7:19:27 AM
View Wi-Fi Info
Recog Results :
Check Time Place sig Score
8:17:56 PM Home 12:05 AM 0.68
8:19:06 PM Home 12:05 AM 0.67
8:20:06 PM Home 12:05 AM 0.71
8:24:02 PM Home 12:05 AM 0.76
8:25:02 PM Home 12:05 AM 0.76
55 minutes to work
Light traffic on 101

(a) *MIT's OIL learns about indoor RF regions from user prompting (active learning), from [173].*

(b) *SensLoc uses WiFi to fingerprint meaningful places without need for geometric location positioning, from [115].*

(c) *Google Now traffic card, showing the automatically inferred place 'work'* [7].

Figure 2.7: *State of the art mobile location positioning and place awareness systems.*

effectively unique IDs without meaning. SensLoc *is* accurate in its inferences, but this is only clear *post hoc* and *not* in real time unless the user has happened to label the place without prompting. Power consumption is reasonable, but not optimal due its reliance on WiFi scanning. The authors do tackle the issue with variable rate WiFi scanning policies however, but there is more room for improvement. These limitations have implications for real world deployment and usability, and forms the basis of our rationale for studying awareness in this dissertation.

Since the ActiveCampus system [81], the value of user feedback and annotation in location positioning has been realised. Bolliger *et al.* have studied the use of well timed prompts to elicit user feedback of indoor location in their RedPin system [29, 30]. Feedback involved users clicking on a map to correct system inferences. Kim *et al.* used daily user surveys in the attempt to annotate automatically classified places, and the Connecto system [17] relied on participants' unprompted self-reports. Montoliu and Gatica-Perez [156] have developed an on-device multi-sensor approach to place recognition which, again, uses participant self-report to elicit meaning of places. Perhaps the most interesting approach, however, is that of Park *et al.* [173], whose OIL system (see Figure 2.7a) attempts indoor location positioning using WiFi signal strength and prompts the user for intervention if signal strengths fall outside a given confidence metric. OIL illustrates how prompting users for feedback and integrating the feedback back into the context learning process can help improve capture and recognition of location points in an unfamiliar environment. The key problems here, however, are: the requirement for devices to continually scan (at approximately $\frac{1}{3}$Hz) for WiFi beacons, which consumes excessive energy; the evaluation was only performed in a single building, which affects the generality of the approach; it does not capture

[7] Image from: `http://www.google.com/landing/now/` (Accessed 2020-10-1212)

place meaning, nor does it attempt to elicit meaning from users; and it does not integrate into commercially available devices. Nevertheless, the general approach – prompting for, and utilising user feedback in the inference process – is interesting and potentially valuable, as the OIL system shows. We feel it is an area that hasn't been fully explored, particularly with mobile devices, and we contribute to the area with our work in Chapters 4 and 5

2.4.4 Activity Recognition with Mobile Devices

Activity recognition is a broad research field which is concerned with the inference of entity – typically human – activity at a given point in time using sensory data. For brevity, we will only focus on activity recognition with mobile devices for this review. Following successful research of activity recognition in smart homes, e.g. [223, 229], researchers began to look at the feasibility of using mobile devices as tools for enabling and performing human activity recognition. Sohn *et al.* [215] used GSM signal traces to infer human mobility, and Anderson and Muller [8] used GSM signal strength fluctuation to infer basic activities such as walking or driving. Although they used existing infrastructure, i.e. GSM cellular networks, these systems could not infer fine-grained activities easily. Eagle and Pentland took a different approach with their Reality Mining project [67]. By capturing Bluetooth data over time and analysing the entropy of users' lives, they could infer basic activities such as 'working' to a good degree of accuracy. However, activities were still coarse-grained. Choudhury *et al.*'s mobile sensing platform [43] – a standalone embedded activity recognition device – addressed this issue at the cost of a custom hardware requirement. It did, however, illustrate the value of fine-grained activity recognition; particularly in relation to health and fitness activities [50, 184]. Choujaa and Dulay [44] used a temporal (rather than location-centric) approach to activity recognition with mobile devices in their TRAcME system. Their approach – which utilised Bluetooth and cellular data captured during the Cityware project [124] – used learned temporal patterns of users' days to classify a set of common activities with good performance.

As accelerometers began to appear on commercial mobile devices, Miluzzo *et al.* [154] used them as part of the CenceMe project to infer 3 basic activities: walking, running and no activity (stationary); and Brezmes *et al.* [32] explored a slightly larger range, where good classification performance was achieved for a set of common activities such as 'walking', 'sitting' and 'falling'. Bieber *et al.* [26] extended this range to include 'driving' and 'cycling', and Yang [243] performed a detailed study that compared a the performance of a range of algorithms for activity recognition with mobile devices. Accelerometers have continued to be the key sensor used for activity recognition on mobile devices [131], with application to health and fitness. Following work by Hong *et al.*

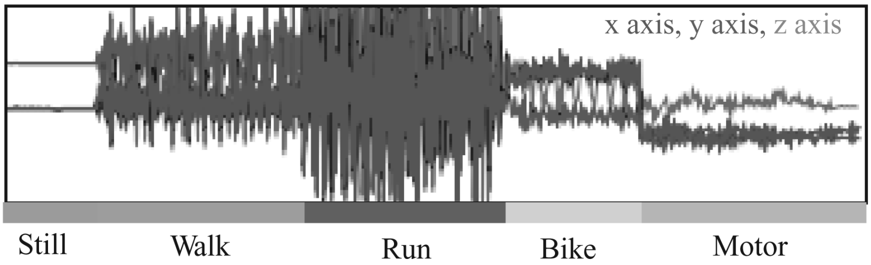


Figure 2.8: *Activity recognition using a mobile device, from Reddy* et al. *[191].*

using an embedded accelerometer [96], Kwapisz, Weiss and Moore [128] demonstrate the potential for real time activity recognition with mobile devices. Figo *et al.* [69] undertook an extensive study that compared alternative activity recognition approaches with mobile device accelerometer data, and Reddy *et al.* [191] used mobile devices to infer transportation activities (see Figure 2.8). This work used fusion of accelerometer and GPS, and a hidden Markov model (HMM) to achieve high classification accuracy in an empirical evaluation. Although these studies show promise in the activity recognition capabilities of mobile device accelerometers, the range of activities recognised is small, e.g. ‘walking’, ‘running’ or ‘sitting’. This is more a limitation of the hardware, as mobile device accelerometers are primarily designed for device-specific interactions such as screen orientation or gaming. The more successful activity recognition studies, e.g. Bao and Intille [16] use multiple, body-worn accelerometers to achieve greater performance. We do not focus on the *classification* of activities in this dissertation for these key reasons: i) previous studies show that only a small range of activities can be reliably classified by mobile devices in practice; and ii) we constrain ourselves to mobile devices and not wearable computing (which is a research field in its own right), thus we wish to avoid the requirement for additional body-worn hardware such as those used in other activity recognition studies, e.g. [16, 92].

A number of studies have used the mobile device accelerometer as an event-based trigger for further sensing or user prompting. Bolliger *et al.* [30] used it to prompt for location annotations, Kjærgaard *et al.* [119] for detecting stop-go motion, and Kim *et al.* [117] for controlling the frequency of WiFi scanning. Ho and Intille, in their work on activity transition detection [92], show that motion-triggered prompts are useful and have potential applications for triggering other processes and applications. However, their work was undertaken using body-worn accelerometers, and not using commercially available mobile devices. The work is relevant, however, as it is not concerned with activity classification – which, as we note above, is non-trivial on mobile devices – but activity *transition*, which is more feasible yet useful for applications such as notification delivery and sensing policy.

2.4.5 Context Inference and Learning with Mobile Devices

Machine learning is a broad topic within the field of artificial intelligence (AI), with a rich and varied history [199]. Once again, for brevity, we will focus upon its application to mobile context awareness. Machine learning is primarily concerned with the automated identification of patterns within data, and the automated classification (or labelling) of data. It has important implications for mobile context awareness, as mobile context aware systems typically perform statistical inference and learning of context given a set of input data. Indeed, many researchers in the field have used machine learning techniques to help improve context inference, or have used mobile context awareness to help improve applications of machine learning.

Lieberman and Selker [138] presented a high level proposal for context aware systems to learn from user behaviour (rather than simply sensing it and reacting to it). Researchers have since implemented a variety of machine learning techniques in order to learn about behaviour. The Technology for Enabling Awareness (TEA) project [78], which ran until late 2000, provided the first in-depth approach to learning in context awareness. Van Laerhoven produced early, detailed work [230] investigating context learning in real time using Kohenen Self-Organising Maps (SOMs) [121]. Van Laerhoven's work presents an interesting perspective on unsupervised learning with mobile devices. Although lacking in empirical study, its theory influences our work on interactive intelligence later in the dissertation.

Using GPS traces of people's daily lives, Ashbrook and Starner [11] applied unsupervised learning techniques to the GPS data in order to extract the meaningful places in people's lives, and Mayrhofer, Radi and Ferscha [151] used a combination of a mobile phone, a PDA and additional sensory hardware to learn about user context patterns. The input data were then used to predict future context based on patterns identified during the learning process. Patterson *et al.* [176] employed sophisticated particle filter techniques with learned parameters in order to classify users' high level behaviours from GPS traces, and showed how activity and location classification performance could be improved using learning. When the Reality Mining results were reported [67], Eagle and Pentland demonstrated how machine learning techniques could be applied to extract temporal patterns in users' daily lives; namely social groups and applications used. In the same year, Horvitz *et al.* released BayesPhone [98], which – unlike the Reality Mining project – implemented pre-learned Bayesian networks of users' context variables on mobile devices. BayesPhone would attempt to estimate users' interruptibility or 'cost of interruption', i.e. whether users should be interrupted by the device given the current context data and the Bayesian network of context variables.

Hightower *et al.* employed a method of 'fingerprint learning' in their BeaconPrint system [90]. This used real time context data to update stored models of places determined by WiFi and GSM base station IDs, and showed that the learning process helped in future inference of users' meaningful places. Nurmi and Koolwaaij [164] took a different approach, which used traditional unsupervised learning (specifically, k-means clustering) to identify places from geometric location data, i.e. GPS coordinates. Their approach was also capable of learning about users' places in real time; as opposed to other approaches that could only learn offline or *post hoc*, e.g. [11, 98, 110, 176]. Krause *et al.*'s work [125] using wearable sensor arrays employed real time machine learning in order to model users' preferences, and Liao, Fox and Kautz [137] applied Conditional Random Fields (CRFs) – discriminative probabilistic graphical models that are used frequently in natural language processing (NLP) – to users' GPS data in order to infer significant places and activities; which improved upon earlier performance using particle filter techniques applied to the same problem [176].

Choujaa and Dulay's TRAcMe [44] used learned models of users' activities to recognise daily activities, and the CenceMe project [154] used multiple on-device audio and activity learned classifiers coupled with offline location and social classifiers to infer users' context through their mobile devices. The key attribute of these approaches (and earlier work on BayesPhone [98]), was the partial implementation of machine learning techniques *on-device*. The feasibility of employing fully on-device learning was beginning to emerge, and Choudhury *et al.*'s mobile sensing platform [43] showed the value of using real time on-device learning for improving context inference and the user experience [50].

The value of using machine learning in mobile context awareness has become increasingly realised, particularly due the diversity of behaviours in mobile device users [68]. Much of the recent work surrounding machine learning in mobile context awareness is concerned with comparing alternative machine learning approaches to context inference problems, e.g. Anagnostopoulos *et. al* [7] compare a range of learning techniques when applied to the problem of location prediction on mobile devices, Luštrek and Kaluža [148] compare alternative learning approaches to fall detection from motion data and Yang [243] compares a set of common supervised learning techniques when applied to mobile device activity recognition. Researchers then began to investigate fully on-device learning and the possible impact it may have on power consumption. Wang *et al.* [233] undertook an extensive study of on-device travel, activity and sound inference using mobile devices, showing that good classification performance could be achieved without significant impact on device battery life.

Miluzzo *et al.*'s Darwin phones [153] (Figure 2.9) use a novel approach to learn about context sensing and inference models entirely from users' mobile devices. By 'evolving' trained classifiers over multiple devices, the authors show how – using a case study of

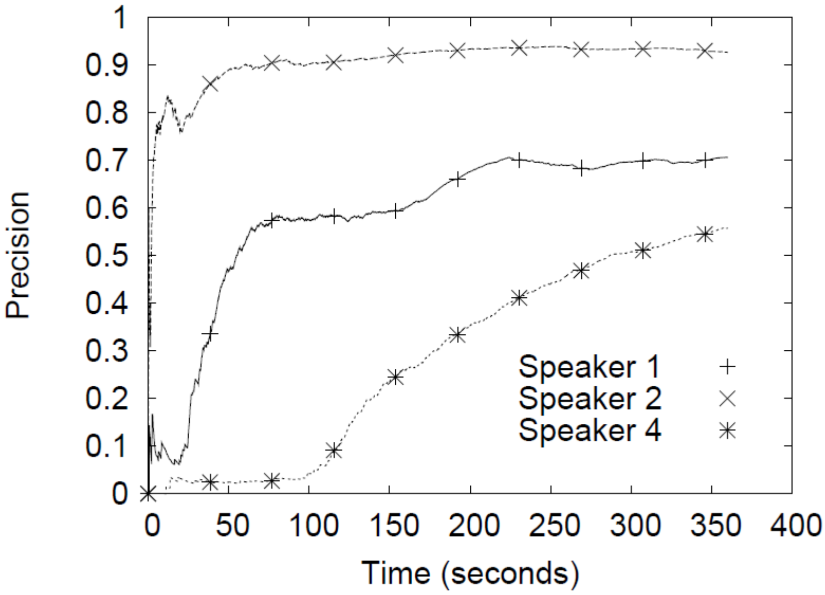


Figure 2.9: *Darwin phones: demonstrating how machine learning improves inference performance. Here, speaker recognition precision (for 3 people) increases as more sound samples are taken in a noisy restaurant. From [153].*

speaker recognition – classification performance can be improved with little need for user input. Choujaa and Dulay [46, 47] use the Reality Mining dataset to show how human activities can be predicted through learning about activity patterns over time; and Lim and Dey [139] compare a range of machine learning approaches to context inference and prediction when attempting to explain automated inference decisions to device users. Rachuri *et al.*'s EmotionSense system [187] is used for the challenging yet novel task of learning and classifying mobile device users' emotions for social psychology research, whilst Sadilek and Kautz [200] use machine learning to infer mobile device users' intent in a game scenario.

There are two key research problems associated with learning users' context with mobile devices: (i) device resources – particularly power – are limited, therefore the application of *useful and usable* machine learning techniques on-device is non-trivial [118, 233]; and (ii) the ideal objective of fully automated context inference through learning is unrealistic, as automated systems will inevitably make incorrect inferences which – in a learning situation – may propagate and affect future inferences [153]. Conversely, we cannot expect mobile device users to diligently correct or update their context without incentive or tangible purpose. The interesting challenge for researchers, therefore, is how we engage the user in the learning process with minimal burden, whilst maintaining or improving context inference performance.

In our work, we are particularly interested in applying machine learning techniques on-device in real time, such that mobile devices are *reactive* to users' context and context changes. Much of the previous work in this section uses pre-learned models, e.g. [98], or *post hoc* pattern classification, e.g. [137], and few have undertaken real-time learning. CenceMe [154] moves this forward, but still relies on offline learning to function fully. In the next section, we look at the feasibility and previous work surrounding a particular type of machine learning – active learning.

Active Learning

Active machine learning is a branch of machine learning that attempts to engage 'oracles' – typically humans – in the machine learning process in order to improve classification performance [49, 210] (see Figure 2.10). Whereas traditional supervised machine learning requires training using relevant data, and unsupervised learning looks for structure in data, active machine learning combines both through intelligent reasoning about the *confidence* or certainty of classification.

Researchers have begun to use forms of active learning in mobile context awareness in order to tackle problem (ii) listed above. The most notable of these is MIT's OIL [173] indoor localisation system, which prompts users for validation of new, uncertain location data when attempting to infer indoor location zones (see Figure 2.7a). Other examples include RedPin [29], Rosenthal, Dey and Veloso's interruptibility learning [196] with mobile devices, Fisher and Simmons' interruptibility learning with reinforcement learning [70] and Kim *et al.*'s SensLoc [115, 117].

These studies are some of the closest to our later work on interactive intelligence on mobile devices, and they showcase the issues involved with eliciting feedback from mobile users in the field, and the advantages of that feedback if it is obtained. Mobile devices are different to desktop systems for interaction behaviour however, and there are very few active learning studies involving them. Hence, we feel it appropriate to incorporate active learning into this dissertation in order to break new ground in this area.

2.4.6 Resource Efficiency

In the previous section, we reviewed work that used machine learning techniques to infer and learn about user context through their mobile devices. As we saw, researchers are often hampered by the resource limitations of mobile devices; mainly battery power and, to some extent, CPU performance and available memory. The limitation of on-device resources is an ongoing problem, and researchers have been tackling the issue by attempting to improve the *resource efficiency* of their context sensing and inference

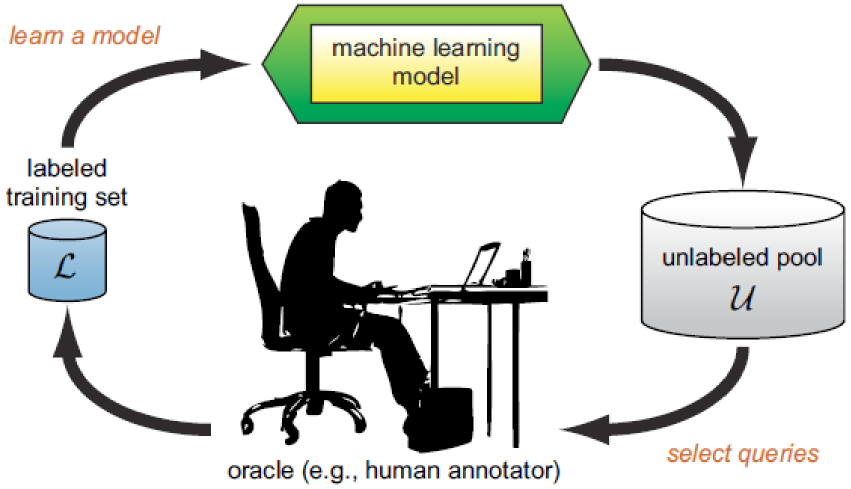


Figure 2.10: *The model for active learning, showing how an automated system can prompt the 'oracle' for confirmation on uncertain classifications. (Image from [210].)*

approaches.

It is a well-known issue that using mobile device sensors significantly affects battery life [118, 233], but researchers frequently require continual sensing for good context inference performance, e.g. [173]. One of the biggest challenges currently faced is knowing when to use costly sensors to capture context data, and when to turn them off to save battery resources – the so-called energy-accuracy trade-off [186]. Various approaches have been taken to address this problem: some systems attempt to balance resource consuming tasks between the device and external servers, e.g. [153, 154, 173]; whereas others try to predict the best opportunity for turning on sensors when needed, e.g. [117, 167, 190, 233].

For state-of-the-art resource efficiency, Li *et al.* [136] implement machine learning techniques to create sensing models that improve the efficiency of context sensing, and Ra *et al.* [183] break up background application tasks in the attempt to optimise energy use on mobile devices.

We do not focus too heavily on resource efficiency in this work. It is an important topic, and one that is well studied in mobile context awareness. Because of this, we do not concentrate on optimising resource usage in our sensing and interactive intelligence work, but we are aware of it and show, later on, that using event-based sensing policies actually improves power usage beyond the current state-of-the-art dynamic polling policies, e.g. [117].

2.4.7 User Interaction with Mobile Context Aware Devices

Awareness

Context aware computing has played a large role in HCI due to the inherent relationship between context aware devices and their users. The most relevant area of HCI to context aware computing is *awareness* [209], i.e. enabling perception – in this case – of context. Context aware computing implies that the context aware device is the entity that is perceiving context, i.e. user context, and many definitions support this, e.g. [57]. However, in traditional HCI awareness, designers are typically concerned with making the *user* aware of events on a computer interface (for example). We therefore have two key areas of awareness: (i) device awareness of users' context, i.e. user → device awareness; and (ii) user awareness of the device interface, i.e. device → user awareness. User → device awareness concerns the device's automated inference of user context – which we have covered in the previous active research sections – but device → user awareness is just as important an issue when designing context aware systems; particularly mobile context aware systems [222].

Peripheral Awareness

Context aware devices may need to make users aware of different things, e.g. the availability of new information, services, applications, or requests for interaction. Often, the key issue for researchers is the approach used to raise users' awareness when the relevant task is not the user's primary task [76, 82, 209]. This has been referred to as "peripheral awareness", which is described by Pederson and Sokoler [180] as "*our ability to maintain and constantly update a sense of our social and physical context*".

Various methods of enabling peripheral awareness have been made in the literature. Weiser and Brown's "Dangling String" [237] project was designed to raise peripheral awareness of network traffic at Xerox PARC by rotating a motor attached to a plastic string in proportion to the amount of network traffic passing through an Ethernet cable. The authors argued that the interpretation of traffic density – which was typically displayed on a screen – was made simpler by abstracting the information into a simple artefact, i.e. the string. Simple sounds are commonly used for raising peripheral awareness of messages, e.g. [201], and have been used for the peripheral awareness of workers' presence and availability [161].

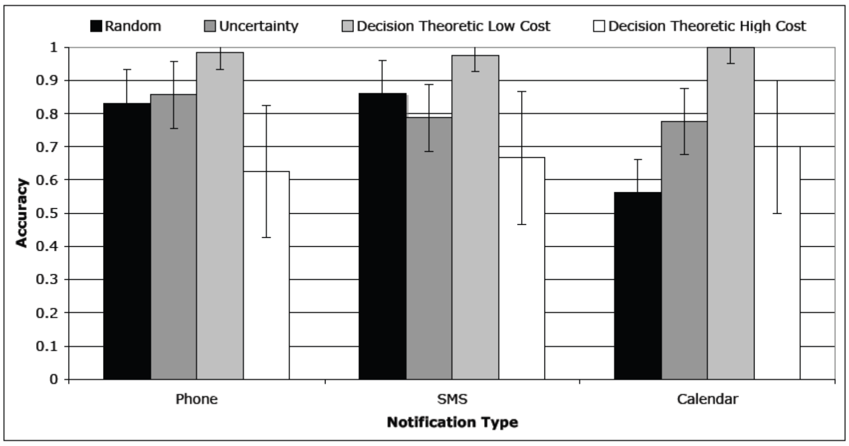


Figure 2.11: *Rosenthal, Dey and Veloso use decision-theoretic decision sampling to construct interruptibility models for mobile devices. Here, the accuracy of interruptibility prediction is measured according to the prompt conditions; high and low cost refer to users' self rated cost of interruption. (Image from [196].)*

Interruptibility

Another important research area in mobile context aware systems is interruptibility. How do we know when users are interruptible? Inferring this is non-trivial (getting it wrong can lead to frustration cf. Microsoft's paper clip assistant), particularly in a mobile environment where users may be carrying their device in different environmental and on-body locations [152]. Using Moran and Dourish's example of a phone ringing (or rather, not ringing) at a concert due to its awareness of its user's context [157], Brown and Randell [33] discuss how such a trivially stated problem can be extremely difficult to implement; due mainly to what Hudson *et al.* [100] refer to as "*a complex tension between wanting to avoid interruption and appreciating its usefulness*".

Fogarty *et al.* have shown that, by using a simple set of sensors (namely: desktop event logs, cameras and microphones), it is possible for context aware systems to predict interruptibility as well as humans can [71, 101]. Knowing *when* to interrupt people is a key issue, and Ho and Intille [92] developed a novel method to identify key points using activity transitions. By deploying an array of body-worn sensors, the authors showed that by prompting users at the point of activity transitions, the perceived burden of interruption was less than if prompted at random. This is a key finding, as it shows that context transitions are potentially useful indicators of user interruptibility. However, the requirement for specialist hardware limits the general applicability of the work, particularly to everyday mobile device users.

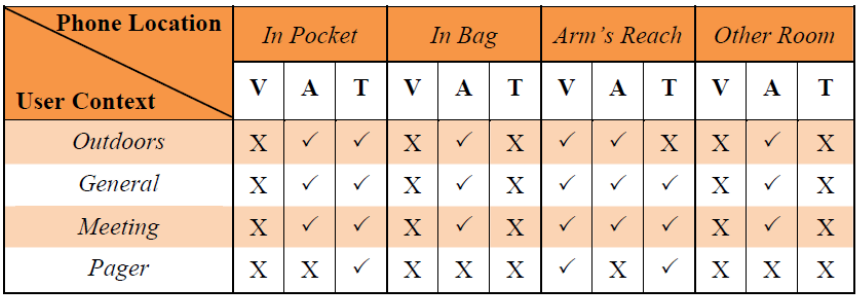
However, recent research has explored how interruptibility can be inferred and even learned using mobile context aware systems. Kern and Schiele [114] show how interruptibility can move from body-worn sensors to mobile devices, and Rosenthal *et al.* [196] show how models of user interruptibility can be learned using mobile devices (see Figure 2.11), which stems from Kapoor and Horvitz's [112] studies of interruptibility learning methods. Rosenthal *et al.*'s work is an example of a learning approach to interruptibility, which is more elegant but requires long periods of daily training, whereas Ho and Intille's work [92] is an example of a reactive approach, i.e. the point of transition is assumed to be a good time to interrupt users, which requires little training but is perhaps more crude in its design. These approaches of predicting interruptibility using mobile devices do however show that the difficulty of automatically inferring interruptibility [33] can be somewhat overcome.

Notification

Even when interruptibility can be inferred, there are still issues surrounding the type of notification used to capture the user's attention. The modality of notification is an important design choice, and it is one that has implications for how receptive users might be to interruption. Hinckley and Horvitz [91] discuss the issues associated with traditional alerting mechanisms on phones, and early mobile context aware systems, e.g. the TEA [78] and SenSay [212] projects, were concerned with lessening the negative impact that notifications have on mobile device users. Hudson *et al.* argued [100], however, that notifications should be designed to be more effective rather than the alternative approach of designing around their ineffectiveness.

Modern smartphones have a range of primary notification modalities: visual (display notifications), audio (ringtones and notification sounds) and tactile (vibration). Figure 2.12 shows Garzonis' map of notification modality effectiveness given user context and device location [75]. As the table shows, audio notifications are generally the most effective, followed by tactile (if the device is on-body) and visual. This links back to peripheral awareness; audio and tactile notifications can alert users without distracting them from their primary tasks, but visual notifications rely on users' partial or full focus.

Garzonis' work using earcons has been of particular influence to us in our later place awareness study. By showing how users associate particular sound patterns with notifications, his work offers novel means of approaching user feedback elicitation from mobile devices in the field.



Phone Location / User Context	*In Pocket*			*In Bag*			*Arm's Reach*			*Other Room*		
	V	**A**	**T**	**V**	**A**	**T**	**V**	**A**	**T**	**V**	**A**	**T**
Outdoors	X	✓	✓	X	✓	X	✓	✓	X	X	✓	X
General	X	✓	✓	X	✓	X	✓	✓	✓	X	✓	X
Meeting	X	✓	✓	X	✓	X	✓	✓	✓	X	✓	X
Pager	X	X	✓	X	X	X	✓	X	✓	X	X	X

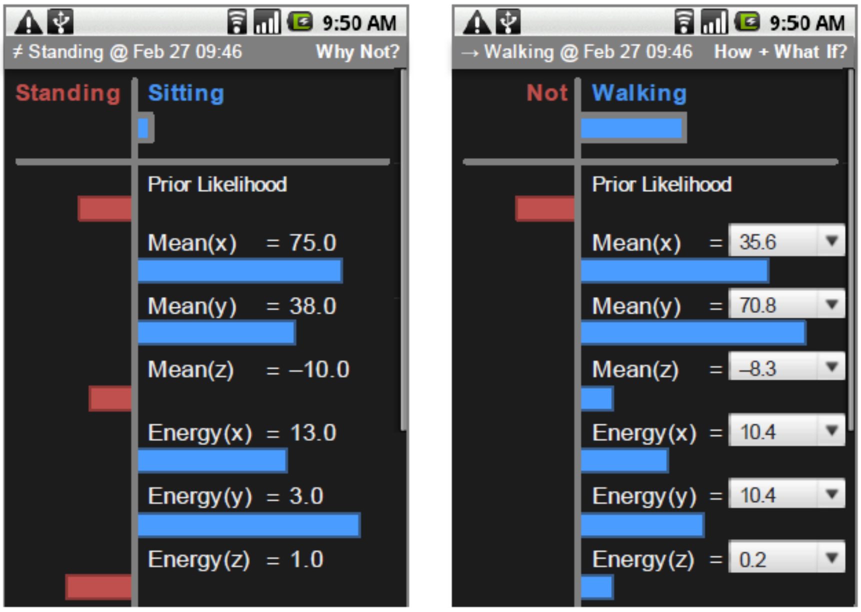
Figure 2.12: *The effectiveness of different mobile device modalities given user context and device location. V = visual; A = auditory; and T = tactile. (From [75].)*

Intelligibility

One of the primary research trends in user interaction design for mobile context aware systems has been *intelligibility*. This is the problem of communicating useful information about the device's state or decision making process to the user so that the user may better understand the reasoning behind the inference [20, 141] and possibly intervene to correct it [210]. User experience may be hampered if the user feels confused or annoyed by the devices automated inference outputs and, without clear understanding as to why the device did what it did, or how to remedy the situation, the user is likely to distrust the application or dismiss it entirely [9, 10].

Much of the recent body of work on intelligibility in mobile devices has been conducted by Lim and Dey. They have used prototype context aware applications to assess users' demand for such intelligibility in context aware applications [141]. The applications include an Instant Messenger (IM) notification plugin; an awareness system that monitors the context of elderly relatives; a reminder system from Dey and Abowd [58] and a mobile tour guide extended from Abowd *et al.*'s early work into context aware computing [1]. The results from this study show that users would be more satisfied if context aware devices were to present a set of 'intelligibility types' to the user – *viz.*: application, situation, input, output, model, why, why not, how, what if, what else, certainty and control – with emphasis on three key elements: *why, certainty and control.*

Lim and Dey go on to extend their work with a toolkit to support intelligibility in context aware applications [139] and early designs of mobile context aware applications with intelligibility integrated within [140, 142] (see Figure 2.13). The predominance in this work is on the 'why' (and, to some extent, the 'why not') explanations for context inference, with initial exploration into the effects of communicating certainty through user interfaces – particularly mobile interfaces.



Mean(x) = 75.0
Mean(y) = 38.0
Mean(z) = –10.0
Energy(x) = 13.0
Energy(y) = 3.0
Energy(z) = 1.0

Mean(x) = 35.6
Mean(y) = 70.8
Mean(z) = –8.3
Energy(x) = 10.4
Energy(y) = 10.4
Energy(z) = 0.2

Figure 2.13: *Lim and Dey demonstrate means of communicating intelligibility to the user in a mobile context aware system; from [139].*

2.4.8 Interactive Intelligent Systems (IIS)

As we saw in the introductory chapter, a recent research field that has emerged from the intersection between AI and HCI is *interactive intelligent systems* (IIS). An IIS is an intelligent system that people interact with [105], and it is the combination of artificial intelligence and human interaction that leads to complex and fascinating research challenges beyond simply telling an intelligent system that it is right or wrong. For example, how much work should be done by the intelligent system *vs* the amount that should be done by the user? Is accuracy a priority, or is it avoiding user burden [219]? How would interaction design choices, e.g. interface types, affect the learning and classification performance of an intelligent system, and how might the classification performance subsequently affect the user experience?

This interesting mixture of HCI and AI problems is especially pertinent for mobile context aware systems. A mobile context aware device is an intelligent system that people interact with, and the vast majority of research in mobile context awareness is either concerned with automated context inference (the intelligence) or the user interaction and experience (the interaction), or both.

Much of the research involved in IIS is new [105], but it is grounded in a variety of other fields. Schmidt first defined the idea of implicit HCI (iHCI) [205, 206], where context awareness could be used to lessen the burden of interaction by removing the more explicit tasks involved in traditional HCI, e.g. direct manipulation of GUIs.

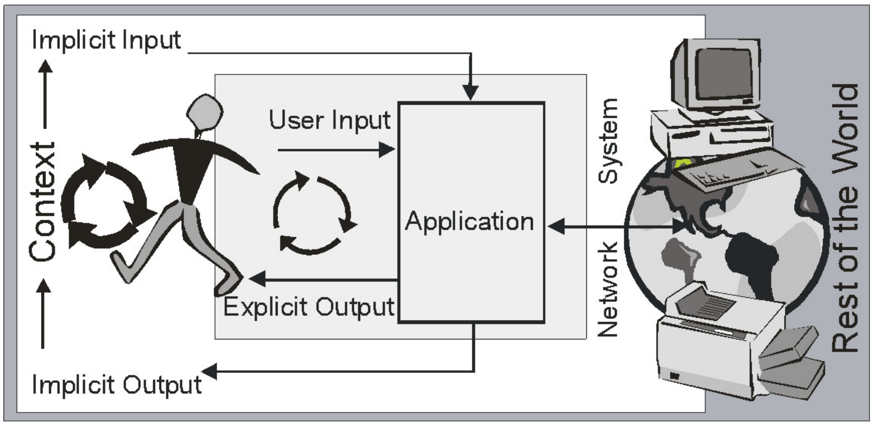


Figure 2.14: *Schmidt's implicit human-computer interaction (iHCI) model, from [206]*

In [206], Schmidt defines iHCI as "*the interaction of a human with the environment and with artefacts which is aimed to accomplish a goal. Within this process the system acquires* implicit input *from the user and may present* implicit output *to the user*" (see Figure 2.14). The "implicit inputs and outputs" are secondary to users' primary tasks or activities, and they allow context aware devices to utilise user input without demanding their direct attention, e.g. turning off a TV when the user has switched focus to a book.

In the ACM's recently launched journal, Transactions on Interactive Intelligent Systems (TiiS) [105], there have been few mobile IISs and, consequently, few studies on how user feedback could be integrated in the context inference and learning process in a mobile environment. Furthermore, there is an interesting question surrounding propagation, i.e. how does inference that utilises user feedback affect future requests for feedback?

In summary, there are two key problems for IIS design that apply to mobile IIS design:

1. How much of the context inference and learning process can we automate without requiring user feedback?

2. How can we elicit user feedback in a mobile environment?

The distinct lack of mobile IIS in the literature is what drives a large part of the research in this dissertation. As we have mentioned previously, mobile devices are very different to desktop systems when it comes to user interaction behaviour and the practicalities involved in deploying machine intelligence. Thus, the normative facets of IIS – the intelligence and the interaction – are likely to be very different for mobile devices. This gap in previous work has allowed us to focus on new research for both intelligence and interaction (and their co-existence) on mobile devices.

2.5 Research Questions

In this section, we narrow the scope of this literature review toward a set of current research problems and questions that this dissertation will address. These are grouped by key active research areas within mobile context awareness.

2.5.1 Context Sensing

As we saw in Section 2.4.1, acquiring context is an ongoing research problem. Designers of context aware systems must first identify potential context sources according to the requirements of their application, before designing sensors that can transduce raw context data from the sources into machine readable form. In the mobile environment, we have an important factor to consider: mobility [131]. By using mobile devices, we can perform context sensing in a naturalistic manner without impacting too heavily on ecological validity due to the need for specialist hardware, but we do have to consider the important and practical constraint of resource consumption, e.g. battery, CPU and data [118].

Even though context sensing is a popular area of mobile context awareness research, there are still ongoing research questions to which we can contribute. First, there is the question of context sourcing and sensing. With the enormous surge of data-driven software services and applications such as social media, and the ever increasing sensing capabilities of mobile devices and computing hardware in general, there lies an interesting question as to what entities we could consider to be context sources and sensors. As we saw, existing approaches to context sensing on mobile devices focus heavily on hardware or 'physical' sensors; usually those present on the devices itself, e.g. GPS and accelerometers [131]. Ground for original contribution lies in 'virtual' sensing [15], e.g. where we can sense context data from a user's Facebook status by using Natural Language Processing (NLP) techniques. Other researchers have considered entities as diverse as biological organisms [127] and photographs [41] as virtual context sensors. This leads us to our first research question:

- **RQ 1**: What entities might we consider as virtual context sensors?

To answer this question, we should identify a set of virtual context sensors and, for a case study, ascribe a performance measure for sensing. By measuring the sensor's performance during the study, we can evaluate whether or not it can be seriously considered as a context sensor.

Although broad, RQ 1 implicitly assumes that context sensors are singular entities, i.e.

a single input (the source) with a single output (the machine readable context data). However, interesting and useful sensing capabilities may be realised by combining or *fusing* sets of context sensors together; which has been previously undertaken in the literature using sets of traditional sensors, e.g. [2, 102, 169]. However, the idea of virtual context sensing leads naturally to the possibility of combining multiple virtual and physical sensors together, with a view to improving sensing performance beyond that of single sensors. If we can identify a candidate sensor for RQ 1 and evaluate its performance, we can also naturally ask whether combining it with other, more traditional context sensors improves overall sensing performance. This motivates our second research question:

- **RQ 2**: To what extent does combining multiple context sensors affect sensing performance?

By addressing these two questions, this dissertation intends to contribute to context sensing in mobile context aware systems. By focusing on virtual context sensors rather than the more traditional physical ones, we will advance upon the current state of the art in mobile context sensing.

2.5.2 Interactive Intelligence: the Intelligence

As we saw in Section 2.4.8, interactive intelligent systems (IIS) have recently emerged in the intersection between AI and HCI. These two fields form the two fundamental components of IIS: the intelligence (AI) and the interaction (HCI) [105]. We will address both of these areas in the scope of mobile context aware systems. For the first – the intelligence – there are two relevant areas of research: context inference and context learning on mobile devices.

Context inference is an extremely broad area and, for mobile context awareness, it encompasses the active research areas that we reviewed in Sections 2.4.1 – 2.4.5. Before addressing inference and learning techniques, one question that frequently arises – particularly in mobile context aware systems – is *when* to turn on or initiate resource-intensive sensors, e.g. GPS, or expensive inference processes [186]: using sensors too frequently is costly if there is nothing worth sensing, e.g. the user's context has not changed; and, conversely, using them too infrequently saves resources but risks missing important events. As Greenberg notes [80], context is extremely dynamic in nature, and knowing when to sense and infer is paramount lest we miss important events or over-consume our available resources.

Current approaches have used periodic sampling [154] and dynamic sampling [186], but Ho and Intille's work [92] using context transition triggers for interruptibility is promising. The authors identify key moments in time when an activity has significantly changed and reason that these moments correspond to moments when users are more likely to respond to interruptions. These context transitions – which Zimmerman *et al.* [246] describe in greater detail – are effectively discrete samples of the continuous dynamic context model [80]. The property in question is the *change* in context over time. Approaches for inferring moments of context change – similar to Ho and Intille's – have been used to initiate mobile sensing policies in the literature, e.g. [30, 117, 119], but these are typically crude, e.g. the use of arbitrary motion trigger thresholds [125] and other parameters [117], and do not take user variability or subjectivity into account [68]. Moreover, there are few formal analyses of transition inference performance in each case; little thought is given to how sensed motion relates to a natural understanding of context transitions, nor how adjusting processing parameters affects transition inference performance. These problems form a basis for our third research question:

- **RQ 3**: To what extent can we infer significant changes in context using mobile devices?

As we discussed in Section 2.4.5, one of the primary relevant research areas is the improvement of context inference and learning performance in mobile context aware systems. The bulk of current research is focused on developing, implementing and improving algorithms that automatically infer and learn users' context from sensor data (see Section 2.4.5), but inferences can still be incorrect due to, e.g. insufficient sensor data or poor quality sensor data.

Automated inference is often seen as an ideal goal because it removes the need for the user to supply the 'answer'. In reality however, it is extremely difficult to make correct automated inferences to the degree of accuracy that users desire [34, 245]. One approach to improving inference performance is to learn about users' context over time using machine learning techniques, e.g. [153], but this still requires some form of user input and too much prompting for such input will likely lead to user irritation [219]. Conversely, too many incorrect inferences are will adversely affect application functionality, also resulting in user irritation. As we saw in Section 2.4.5, an interesting approach to this problem is active machine learning [210]; where an intelligent system can measure the confidence of its own decisions and ask the user for help only when required. This approach is potentially useful as it could avoid the pitfalls associated with the aforementioned automation-supervision trade-off. Furthermore, it could allow us to capture additional data from users – such as intent or *meaning* – that could otherwise be non-trivial to sense and infer.

As we saw, only a few studies have used active learning in mobile context aware systems. Kim *et al.* propose it for future work in their study of people's meaningful places, but do not implement it [115], and Park *et al.* [173] use it for indoor location positioning, but do not study it beyond a single building with specialist devices. The key problem for researchers is the design of interactive intelligence that will perform well in response to *expected* user interaction, not the ideal – we cannot expect users to be 'perfect oracles' [218]; in reality they will likely miss prompts or just ignore them, especially in a mobile environment. Thus, given the potential value of active learning for improving context inference, and given its lack of implementation in mobile context aware systems, we can outline another research question:

- **RQ 4**: To what extent can we infer and actively learn about context using mobile devices?

So, for the intelligence component of interactive intelligence in mobile context aware systems, we concentrate on: inferring moments of context change, and the inference and learning of context itself.

2.5.3 Interactive Intelligence: the Interaction

The other component of interactive intelligence is the user interaction. As Bellotti and Edwards note [20], "There are human aspects of context that cannot be sensed or even inferred, so context aware systems cannot be designed simply to act on our behalf". Given that users can interact with the intelligent system and provide feedback for active context learning, we should also study *when* and *how* we should attempt to elicit such feedback from users if we need it. This is perhaps a more complicated issue than it first appears: prompting users for feedback (if necessary) at the right time is imperative for useful response [34, 140], but when is the 'right' time to prompt? There is then the question of the mode or modes used for prompting, as well as design choices for each, e.g. if we use audio prompts, then the choice of whether to use a simple sound or speech might affect users' feedback response behaviour [75]. Our objective is to elicit context feedback from users for active learning, which may prove non-trivial in a mobile environment where users are unlikely to notice or respond to requests for feedback. This is our penultimate research question:

- **RQ 5**: How can we elicit context feedback from users in a mobile environment?

Given the preceding research question, we should also ask how users typically interact with mobile IISs. As Stumpf *et al.* noted during their studies of desktop IISs [219, 218, 220], there is little research into how end users interact with intelligent systems; rather, most work assumes – much like the field of active learning – that the user is simply an ‘oracle’ that will tell the intelligent system that it is right or wrong on request [210]. Given mobile IISs, where user interaction may be complex due to the extremely dynamic nature of mobile users’ context, this question becomes even more pertinent. Users are subjected to all kinds of external stimuli, and their mobile device interaction behaviour can vary significantly [68]. This question of user interaction in interactive intelligence forms our final research question:

- **RQ 6**: How do users interact with an interactive intelligent mobile context aware system?

It would be useful, therefore, to observe typical user interaction behaviour, e.g. feedback request response rate and time, as well as users’ opinions on the experience of interacting with a mobile IIS in the field.

2.5.4 Low-level Research Questions

Although the research questions presented here form the guide for the work in this dissertation, they are high level and, as such, our work will focus on lower level instantiations of each overarching question. Within each chapter, a set of lower level research questions related to the relevant high-level question will be presented. Of course this immediately limits the generalisation of some of the work, but we do this in order to produce tangible research contributions that can enable further research within the scope of the higher level questions.

To address this further, each study and chapter discussion will assess the implications of the work in relation to the higher level research questions presented here. There will be particular focus on the limitations and what still needs to be addressed in light of the findings. Finally, in the concluding chapter, we will discuss this in the broader context of the dissertation as a whole, and link the outstanding needs to future research agendas involving the high-level research questions.

In the next section, we present our layer model that we will use to structure these research questions in a meaningful and intuitive way. We will first review similar models and their uses in the literature, before presenting ours along with a summary of the chapter.

2.6 Research Structure

In this section, we outline a layer model for the research in this dissertation. As we shall see, layer models are ubiquitous in the literature due to their simplicity, extensibility and the intuitive manner in which they represent abstract concepts; and we conjecture that a layer model is an appropriate choice of structure for our work.

We will first outline various examples of layer models used in existing context aware and mobile context systems, before describing our layer model and where our research questions lie within the model.

2.6.1 Layer Models in the Literature

Using a layered structure to model systems is certainly nothing new. Perhaps the most well known examples are the OSI 7-layer and TCP/IP models used in computer networks [31] to capture the abstractions and dependencies of remote communication between applications in a network. Layering is typically used to separate systems into well-defined components – or modules – with application programmable interfaces (APIs) separating each layer. This separation allows each layer to manage a particular system function, and it removes all but the key dependencies between each layer.

In context aware computing, layer models are a popular choice for illustrating system designs, categorising functions and capturing the important functions that enable context awareness.

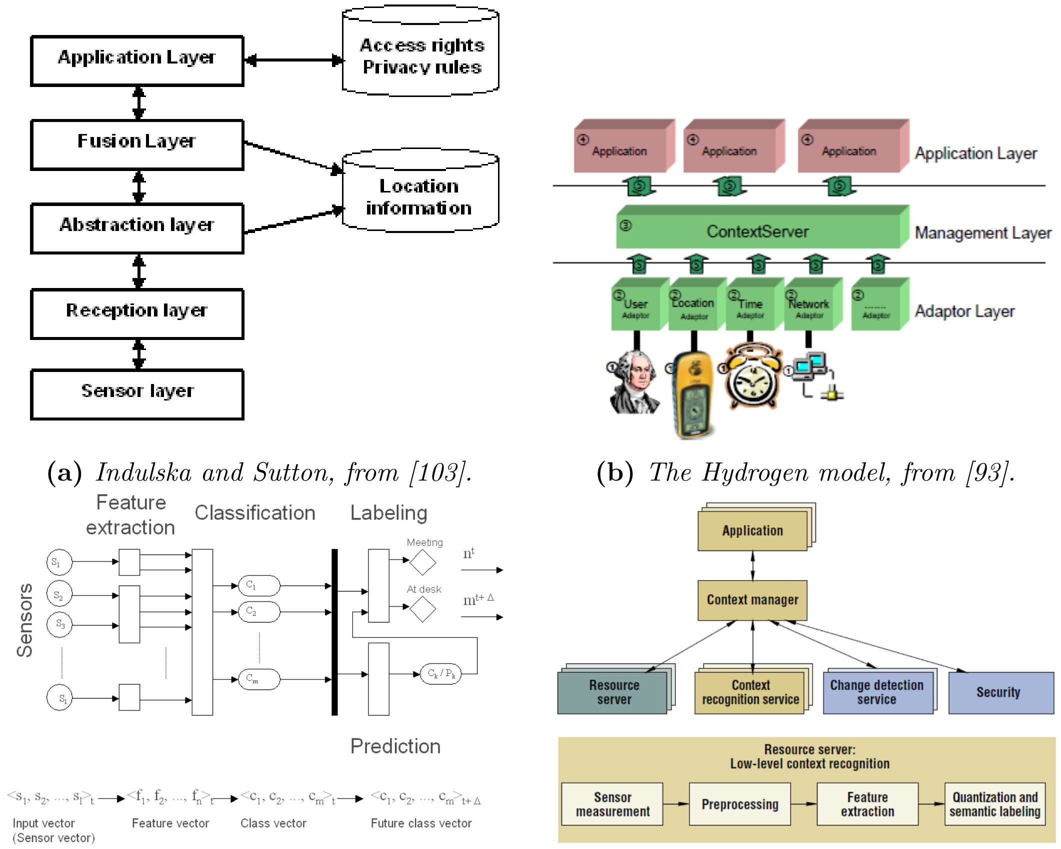
- Indulska and Sutton [103] (Figure 2.15a) use a five layer approach to manage location in pervasive systems.

- Hydrogen [93] (Figure 2.15b) was one the first approaches to use a layer model for mobile context awareness. The lowest layer – the Adapter Layer – was responsible for context sensing and low level data processing, whilst the middle layer – the Management Layer – acts as an interface between the Adapter Layer and the applications present in the Application Layer. Interestingly, the Management Layer does not appear to perform any inference; rather, it simply routes application requests for context data and returns the data appropriately.

- Mayrhofer [150, 151] (Figure 2.15c) uses a layered approach to outline, in great detail, their context sensing and inference processes for context prediction.

- Korpipää *et al.* [123] (Figure 2.15d) specifically model context information flow in mobile devices using a layered approach. Unlike other approaches, there is no ‘sensing’ layer; rather, the sensing functions are contained within a single module on the same level as the inference functions.

- The SOCAM architecture [83] (Figure 2.16a) takes a service-oriented approach, with a clear focus on formal ontology and reasoning. Again, three layers are



(a) *Indulska and Sutton, from [103].*

(b) *The Hydrogen model, from [93].*

(c) *Mayrhofer's prediction architecture, from [150, 151].*

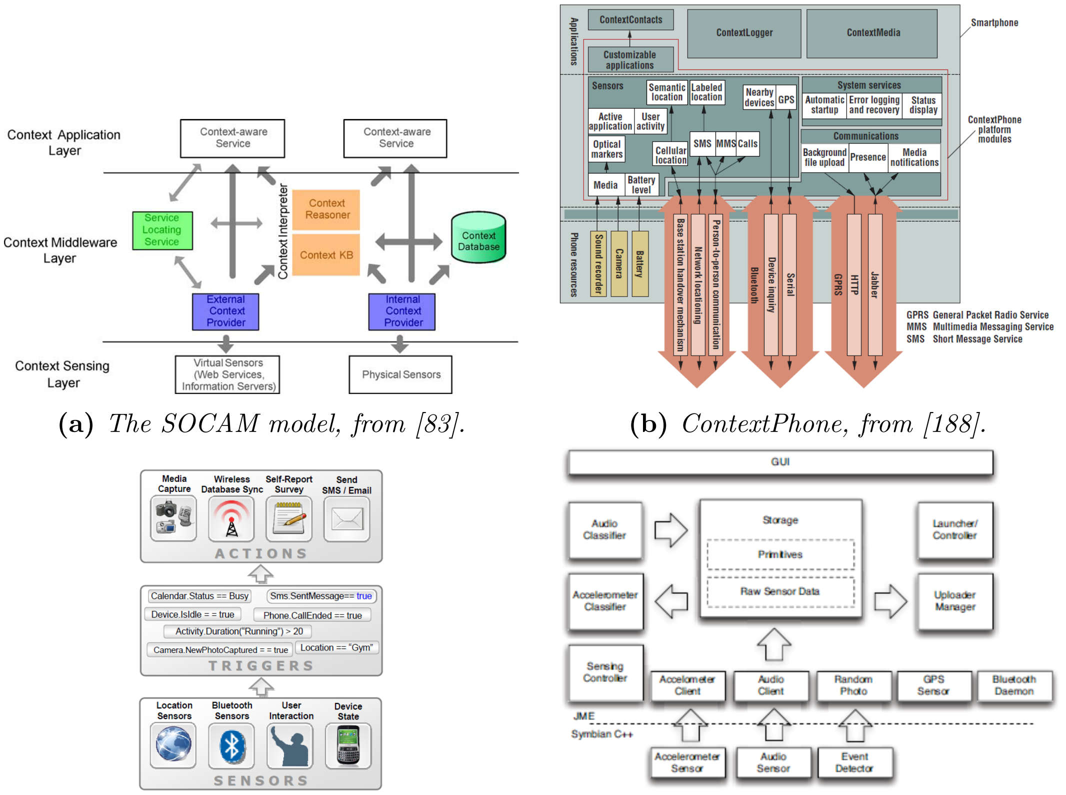
(d) *Korpipää* et al.*'s mobile context awareness model, from [123].*

Figure 2.15: *Layer structures in context aware systems.*

used: context sensing, which extracts data from physical and virtual sensors; context middleware, which performs context reasoning, or inference; and context application, which contains the authors' proposed services. Unlike many of the models listed here, the SOCAM model is very abstract, i.e. the authors use it to structure functions rather than concrete system designs.

- ContextPhone [188] (Figure 2.16b) used a more concrete layer architecture based around its implementation on the Nokia Series 60 (S60) platform for the Symbian operating system. ContextPhone was designed to enable the rapid development of mobile context aware applications by standardising the general structure of a mobile context aware system. Again, three layers are used, but there is a clear focus on network-acquired context data; which forms the majority of the lowest layer. Interestingly, the authors place context sensing in the middle layer, along with services and communication processes.

- Baldauf *et al.* [15] – in their survey of context aware systems – note the common layered approach to context aware system design; also noted by Hong *et al.* in a



(a) *The SOCAM model, from [83].*

(b) *ContextPhone, from [188].*

(c) *MyExperience, from [73].*

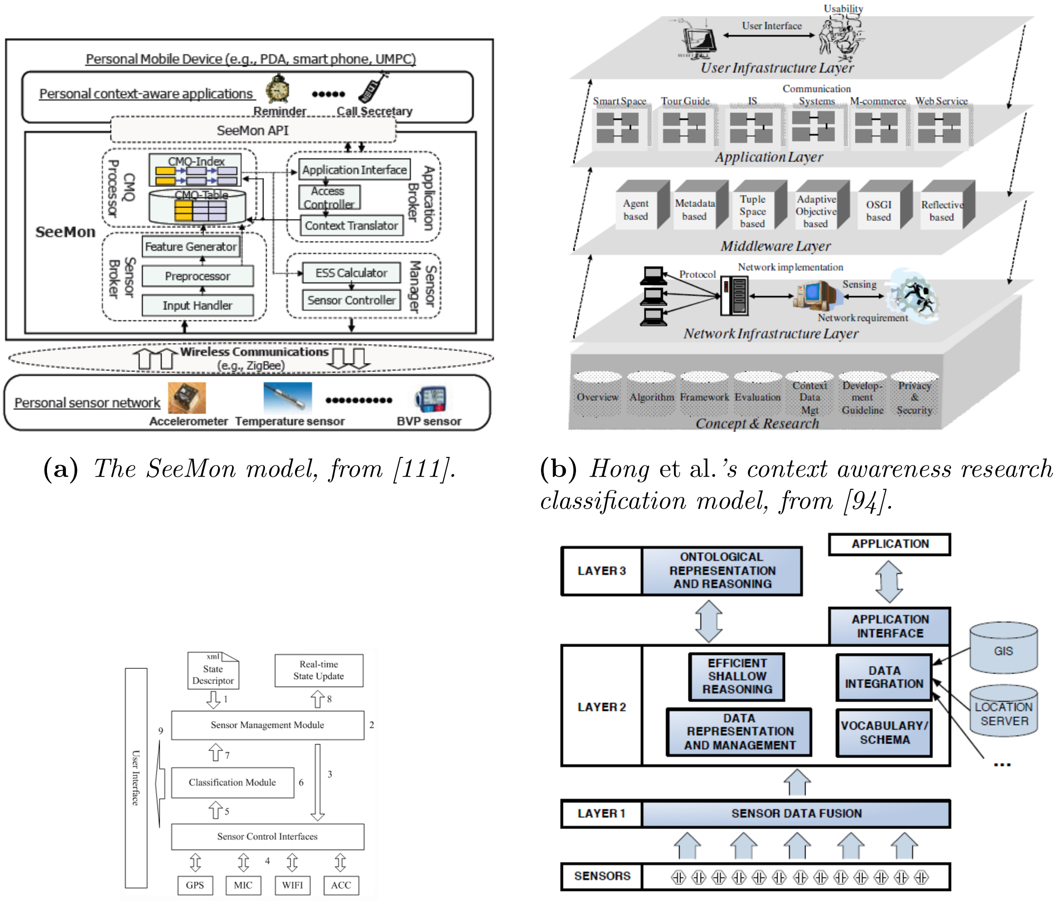
(d) *CenceMe on-device design model, from [154].*

Figure 2.16: *Layer structures in context aware systems.*

later survey [94] and Figo *et al.* in their work on context inference [69]. In fact, the authors use the layer model as a guide throughout their survey, particularly when discussing the key areas of ongoing research.

- MyExperience [73] (Figure 2.16c) uses a three layer approach to capture user experience data in the field using mobile devices. Here, the authors use a sensing layer, but the higher layers are responsible for 'triggers' – which are executed upon interesting sensor events – and 'actions' which are controlled by the triggering layer. This approach allows basic sensor events to drive data logging processes and user prompts, with well defined interfaces between each.

- Miluzzo *et al.* [154] (Figure 2.16d) uses a semi-layered approach to modelling both the on-device and server components of the CenceMe system. Here, once again, the sensing layer is explicitly defined (along with a graphical user interface (GUI) layer at the top of the on-device model). The middle layer, however, is further fragmented into modules with particular inference and storage functions. CencMe is interesting as it uses two separate models for its client-server components; somewhat emulating the TCP/IP layer model for computer networking.



(a) *The SeeMon model, from [111].*

(b) *Hong* et al.*'s context awareness research classification model, from [94].*

(c) *Wang* et al.*, from [233].*

(d) *Bettini's model, derived from modelling survey, from [25].*

Figure 2.17: *Layer structures in context aware systems.*

- SeeMon [111] (Figure 2.17a) uses a classic sensing-inference-application layer model, but the interface between the sensing and inference layers is explicitly defined to be wireless.

- Hong *et al.* [94] (Figure 2.17b) use a layered approach to classify the core areas of context awareness research in their extensive literature survey. By using the layered structure, the authors have managed to quantify the amount of research that has been output according to levels of abstraction.

- Wang *et al.* [233] (Figure 2.17c) use the layered approach to model the design of their energy-efficient context sensing system. Like CenceMe [154], the inference layer is broken into a series of modules that output their decisions to a user interface.

- Bettini *et al.* (Figure 2.17d), in their extensive review of context modelling [25], recommend a layered approach for grouping context sensing, reasoning and application components of a context aware system. The authors cite a series of requirements for rich context modelling, including: sensor heterogeneity, timeliness, dealing with imperfection, context reasoning and usability. The requirements are used to derive the layer structure through an analysis of the context modelling literature, and this layer structure is later used in the authors' COSAR system [193].

Clearly, using a layer model to structure both context aware systems and research helps researchers during the design and implementation of their systems. Many of the aforementioned layer models tend to have some form of user interaction layer at the top of the stack, e.g. [154, 233], but the direction of data flow is typically 'one way', i.e. from the low to high towards the application and user interaction layers. Thus, one of the active research areas not captured by these approaches is user feedback, i.e. directing data flow back into the inference layer.

We therefore use a modified version of the traditional layer model to structure the research contributions in this dissertation. The key difference between our approach and existing ones lies in the addition of a user feedback interface, which allows clean design of user intervention components in both the user interaction and context inference layers.

The next section presents our layer model, summarises each of the key layers and maps the relevant dissertation chapters to the layers.

2.6.2 Our Layer Model

As we saw in the previous section, the number of layers in each architecture can vary between systems. However, descriptions of low, middle and high levels – which imply a generally increasing level of abstractness – appear frequently in the descriptions of these systems. This 'three-layer' approach encapsulates all things 'low level', e.g. hardware, electronics, physics and sensors; 'high level', e.g. applications, user interfaces, behaviours and environments; and 'everything in between', e.g. software processes, algorithms, middleware, data storage, data management, protocols and services.

Like many others, we adopt this approach to structure our research. At the lowest level – sensing – context data is sourced and sensed, and perhaps preprocessed. At the middle level – intelligence – the sensed context data is processed in order to draw conclusions about context. At the highest level – interaction – there is an interface between system users and applications. Furthermore, between the intelligence and interaction layers, there is feedback interface (or 'sub-layer'), which feeds data back into the intelligence layer from the user (through the interaction layer). The layer

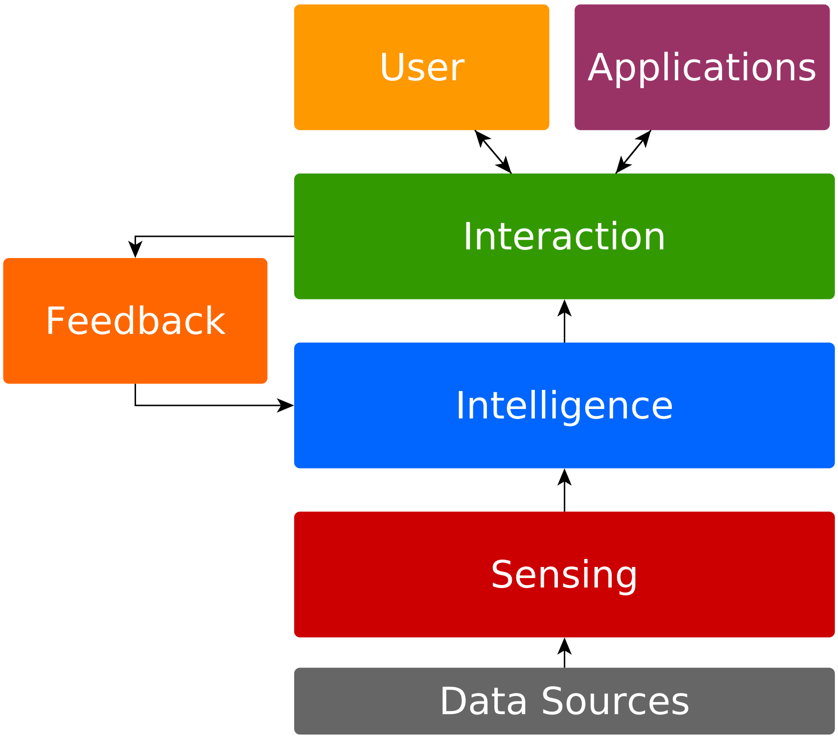


Figure 2.18: *Our layer model for structuring our research.*

structure is shown in Figure 2.18.

Sensing Layer

'Sensing' context can be quite abstract. We have to identify potential sources of context and, depending on the system requirements, define an appropriate set of sensors to convert raw context data into machine-readable data. Traditionally, sensors are transducers of physical phenomena into electrical signals, but the physical definition has been stretched to include non-physical sensors, i.e. 'virtual' sensors, which can 'sense' context data from virtual sources such as software applications or network APIs [15, 83, 103].

In Chapter 3, we begin at the sensing layer to address RQs 1 and 2, which are concerned with what entities could be considered as context sensors, and how combining sensors affects sensing performance.

Intelligence Layer

Inferring and learning about context can often be non-trivial, especially in a mobile environment where sensors may be rudimentary, resources may be restricted and data quality may vary, e.g. RF signal strength varying with mobility. As such, the use of AI and machine learning techniques form a large portion of mobile context awareness research. This is the intelligence component of interactive intelligence. Because intelligence processes are tasked with reasoning, they must accept observations, i.e. data, as inputs and output conclusions based on the observations and, possibly, prior knowledge. This fits naturally with the layering ethos [31], where data from the sensing layer can be input into an intelligence layer which, in turn, outputs conclusions to a higher layer.

In Chapter 4, we use an intelligence layer to address RQs 3 and 4, with the novel addition of input from a user feedback interface. We first consider the problem of inferring significant transitions between context states within a mobile environment (RQ 3), before moving on to the inference and learning of the context states themselves using active learning techniques (RQ4).

Interaction Layer

User interaction – the HCI component of interactive intelligence – is equally as important as the intelligence component in mobile context aware systems. The functional requirement of many context aware systems is to provide or enable a service for a set of end users, so the point of interaction between the end users (or applications) and the system is a critical design component. Again, this fits naturally into a layer structure above the intelligence layer, as we can design user/application interfaces whose only dependence upon the intelligence layer is through the APIs between the interaction and intelligence layers.

It is at the interaction layer that we design the user interfaces for user interaction with mobile context aware systems. In Chapter 5, we tackle RQs 5 and 6 through the interaction layer.

Feedback Interface

One of the key areas that we focus on in Chapters 4 and 5 is the incorporation and elicitation of user feedback in a mobile IIS. In our layer structure in Figure 2.18, we describe a feedback interface *from* the interaction layer *to* the intelligence layer, which allows users to intervene and provide feedback on the context inferences; which is

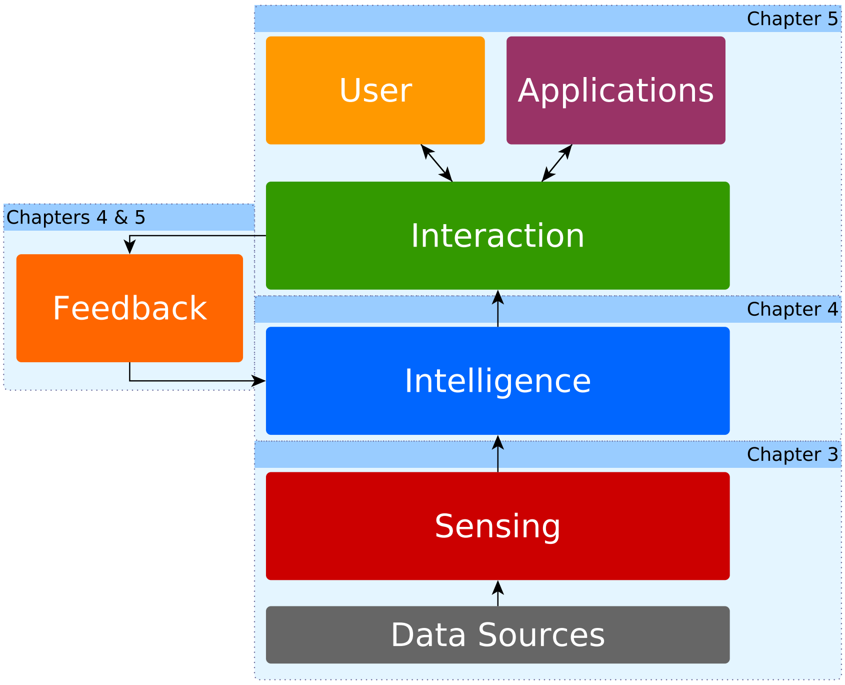


Figure 2.19: *Mapping dissertation chapters to layer model components.*

effectively active machine learning.

We investigate the effects of user feedback on the inference process in Chapter 4 as part of our contribution to RQ 4, but we consider the interface design in greater detail in Chapter 5 as part of RQs 5 and 6. The definition of this interface is crucial when considering the design of mobile context aware systems that utilise user feedback and, because of this, we outline a set of requirements for designers to consider.

Users and Applications

The final components of our layer structure in Figure 2.18 are the users and applications that will interact with our mobile context aware system. Their interactions are made through the interaction layer, which contains the user and application interfaces of the system.

2.7 Model or Architecture?

As we have seen in this chapter, layer models have been used as both representations of context aware systems, e.g. [154], and more formal software architectures, e.g. [188]. There is a question therefore, as to whether our layer model should be developed into a software architecture during this dissertation. As we are using as a guide to our research structure, its development as an architecture will not be the central theme of the work. However, we will continually use it during the designs of our studies; especially to map the modular components of our place awareness system in Chapter 5. In doing this, we show how the model could be developed into a more formal architecture as an agenda for future work. By demonstrating its use in our research, we illustrate its potential as an architecture with formal APIs. In the concluding chapter of the dissertation, we will critically evaluate how the layer model's use in our work has strengthened its case for further development as a formal software architecture for developing mobile context aware systems with interactive intelligence.

2.8 Conclusion and Chapter Summary

In this chapter, we reviewed the literature surrounding context awareness and mobile context awareness. We introduced the key background work that has helped to shape and influence active research into mobile context awareness, including the research areas specifically related to mobile context systems, namely: context sensing, location positioning, activity recognition, machine learning, resource efficiency, user interaction and – importantly – interactive intelligent systems (IIS). The current state of the art was outlined within each, and this sets the boundary for our research questions.

The scope, problems and high level research questions for the dissertation were then derived, and these will be addressed in Chapters 3 – 5. In the next chapter, we tackle our first research questions related to sensing in mobile context aware systems, before moving on to intelligence and interaction – the two fundamental component of interactive intelligence – in later chapters. Throughout this dissertation, our layer structure in Figure 2.18 will be used to guide the research and illustrate how our work fits together.

Chapter 3

Context Sensing

In Chapter 2, we discussed the widespread use of layered approaches when designing, modelling and classifying context aware systems. The lowest layer in the majority of these architectures – usually referred to as a ‘sensing’ or ‘hardware’ layer – typically addresses the task of acquiring or transducing physical data from the world into machine readable data that can be processed by an ‘inference’ or ‘middleware’ layer. The sensing layer is an important interface between the context aware system and the outside world, and it forms the foundation for many context aware systems.

In this chapter, we consider the design and implementation of the sensing layer in mobile context aware systems. We will be addressing RQ 1 and RQ 2:

- **RQ 1**: What entities might we consider as virtual context sensors?
- **RQ 2**: To what extent does combining multiple context sensors affect sensing performance?

For RQ 1, we describe the requirement for context sensing, before analysing a range of context sources present in people’s everyday lives; particularly sources related to the people themselves. In Section 3.2 we link context sources to context facets, and describe three important categories of source: users, devices and the environment in which both operate. In Section 3.3, we provide an overview of context sensors, with focus on the sensing capabilities of mobile devices. We then explain the distinction between ‘physical’ and ‘virtual’ sensors – a distinction which has arisen in the literature [15, 103].

For RQ 2, we discuss how context data can be sensed and combined through low level data fusion; a process that merges data from multiple context sensors in the attempt to improve sensor accuracy beyond that which is achievable by the individual sensors alone.

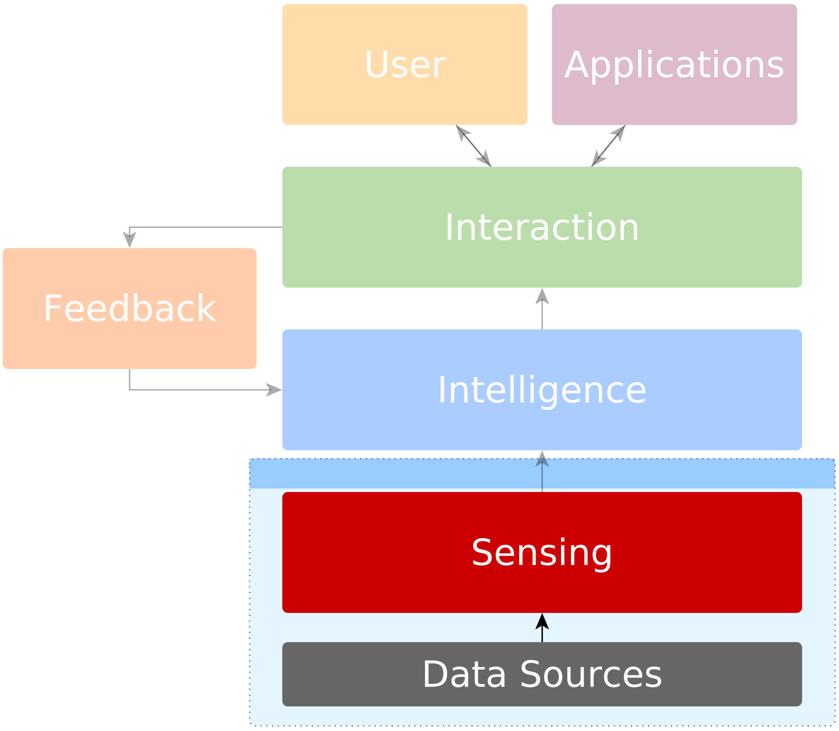


Figure 3.1: *Our layer model, highlighting data sources and the sensing layer; both of which this chapter will be focusing on.*

In order to produce a tangible contribution to these research questions, we present a case study of mobile context sensing in the field; in which we consider the calendar as a virtual context sensor. We develop a set of low level research questions for the study, and analyse the calendars of two software engineering teams in an office environment; showing that the standalone office calendar is a poor context sensor due to the prevalence of ‘noise’ caused by reminders and placeholder events that do not actually take place. Furthermore, by sensing context with other sensors – Bluetooth (location) and the workers’ email address directories (social network) – we design and evaluate two data fusion algorithms that combine the calendar with these other sensors. We show that, in our study, both algorithms significantly improve context sensing performance beyond the standalone calendar; a finding which has implications for context aware presence and availability systems in office environments. Finally, we present a small range of prototype applications that use calendar-based context sensing to provide services to mobile device users in an office environment.

3.1 The Context Sensing Requirement

As Dey's abstract definition states [57], context is *information* about a given situation. To be context aware, a system must process raw context data into useful context information. Although this process is somewhat loosely defined (and often non-trivial), the first step is clear: we must obtain raw context data from *somewhere.* This is the key requirement for a context *sensor* – to translate data from one or more context *sources* into machine readable data about context *facets.*

For most mobile context aware systems, the specific sensing requirements and constraints will depend on the application under development, but the abstract requirement for context sensing remains the same: *to translate raw data from context sources into machine readable data relating to appropriate context facets.* Once sensed, the context data can be passed to higher level processes for inferring context information. There are a number of questions that should be considered when designing the sensing component of a mobile context aware system:

- Which facets of context do we need? Depending on the application, we may only need data relating to a single context facet, e.g. location for the "Where" facet. However, more complex applications may demand data relating to further facets, e.g. activity for "What" or identity for "Who".
- From what sources can we usefully sense context? Once we have identified a set of context facets, we should identify a set of data sources from which it may be possible to sense context from. For example, if we wish to sense user activity, we should identify physical sources of activity, e.g. the user's body movement.
- What sensors are available to us? Having identified a set of potential context sources, what sensors can we use to obtain our machine readable data? We may need to design and implement our own sensing system, or we could utilise existing infrastructure. Again, this is application dependent, but the choice of sensors will have an impact on the data quality and implementation costs the system.
- Are there any constraints? We may have cost or availability constraints for sensors, e.g. we may choose a low quality sensor such as a mobile device accelerometer to sense user activity data due to the ubiquity of mobile devices; this ubiquity has advantages for large scale, rapid deployment and some guarantee of hardware standardisation. There may also be legislation and privacy constraints, e.g. even if it is possible to sense certain context data, we may be restricted in our storage and usage of it.

With these design questions in mind, we outline a range of context sources and sensors before describing the differences between ‘physical’ and ‘virtual’ context sensors. We will link examples of sensors to their context sources and facets, before reporting on our field study of the calendar as a context sensor.

3.2 Context Sources

In this section, we outline a set of context sources which, in these cases, are entities that generate context data.

3.2.1 People

The primary entities from which from which we can source context data are people [57, 202]. People are an important source of data for each of the key context facets as they are the users of context aware systems. People have to be somewhere, they have to be doing something at any given time, and they may also be doing these things with other people.

3.2.2 Devices

The secondary entities that can generate context data are electronic devices. In the mobile domain especially, devices are effectively a conduit to the users themselves. For example, when we attempt to sense a user’s location, we are really sensing the device’s location and making the tacit assumption that it also the user’s location. Indeed, it is within the devices that many context sensors are contained and, although we may assume devices and users are usually in close proximity to each other, we must be aware that we are not always sensing data from the user directly [175].

3.2.3 Environment

The peripheral entity from which we can source context data is the environment in which both the user and the device are operating in. Typical data generated by the environment, e.g. temperature, humidity, pressure, luminosity and ambient noise, can be useful forms of context data.

3.3 Context Sensors

Once we have identified the key context sources for our system, we need to determine sensors that could be used to translate raw data from the context source into machine readable data about relevant context facets. We constrain our overview here to sensors that are readily available in most modern mobile environments, i.e. sensors that can be accessed by mobile devices either locally (on-device) or remotely (through a network), rather than proprietary sensors that may be embedded in existing infrastructure and are difficult or impossible for most mobile devices to access, e.g. CCTV cameras. For our overview of mobile context sensors, we use an important distinction between two sensor types – physical sensors and virtual sensors [15, 103].

3.3.1 Physical Sensors

Physical context sensors are hardware sensors that are used to sense data about people, devices and environments through a *physical* interface; they are explicitly designed as transducers of physical properties. Figure 3.2 illustrates the relationships between facets, sources and example physical sensors available to most modern mobile devices.

3.3.2 Virtual Sensors

Virtual sensors are software sensors that are used to sense context data about people, devices and environments though a *virtual* interface. Such interfaces are either local or remote application programmable interfaces (APIs) which can be queried for context data, e.g. temperature from a weather website API rather than a physical thermometer. Many modern websites – particularly social media sites such as Facebook or Twitter – can be seen as virtual sensors; by using their publicly accessible APIs, mobile devices can access swathes of data relating to people, devices and environments. One of the main advantages of virtual sensing in a mobile environment is the potential for reducing resource use, e.g. rather than operating an on-device physical sensor – which is likely to consume a reasonably large amount of power – a device could simply query an API. This of course implies a connectivity cost, but in the modern smartphone era, where data is relatively cheap and battery technology is not advancing as rapidly as smartphone hardware [118], the cost of power consumption is potentially more valuable to the user than the cost of data.

Figure 3.3 illustrates a range of virtual sensors and the relationships between context sources and facets. Many virtual sensors can be accessed locally by a mobile device, e.g. a contacts address book, but the 'always on' nature of smartphones, coupled with

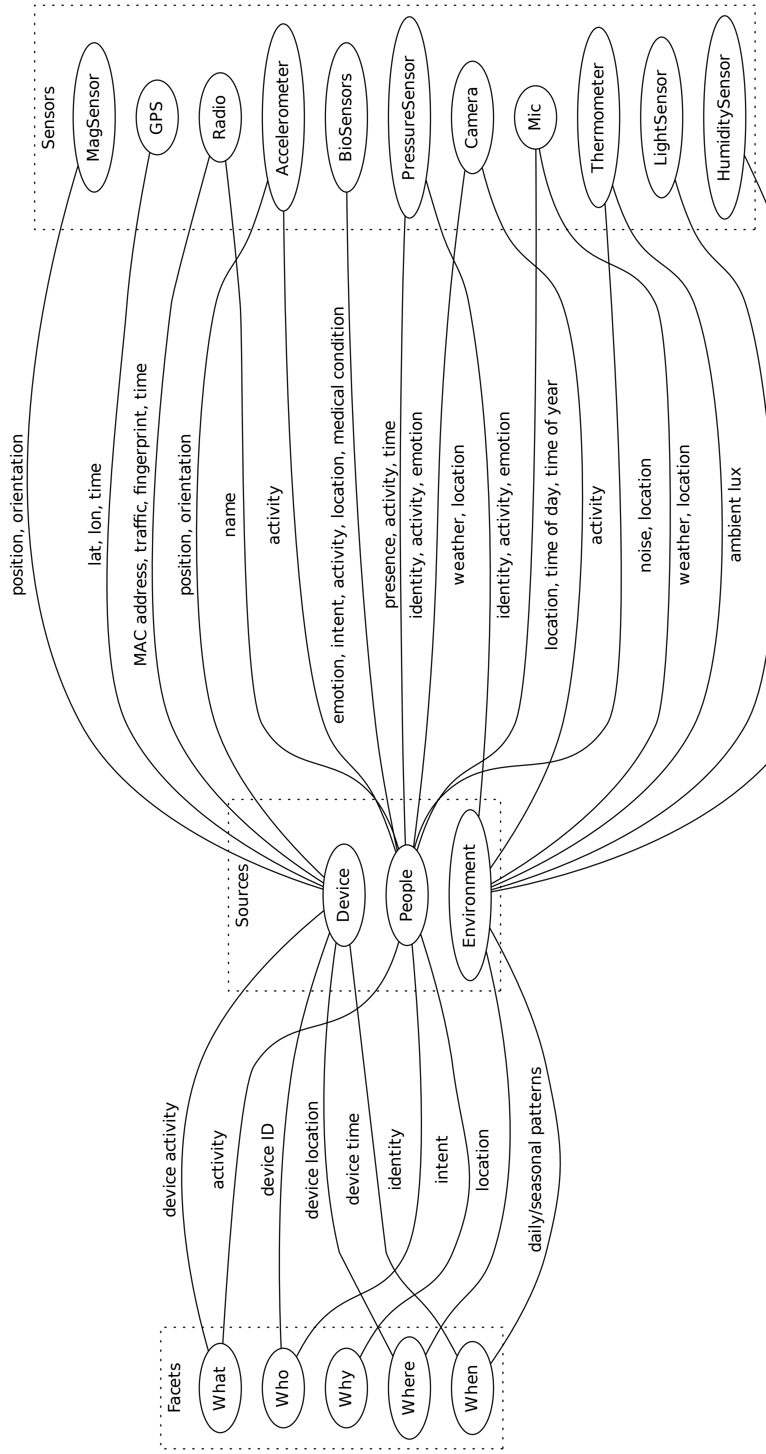


Figure 3.2: *Examples of physical context sensors, with associated sources and facets. Edge labels from source to sensor show example instances, i.e. the relationship should be read as "<sensor> senses <edge> about <source>", e.g. "mic senses identity about people". Edge labels from facet to source should be read as "<source> provides data about <facet> through <edge>", e.g. "people provide data about 'who' through identity".*

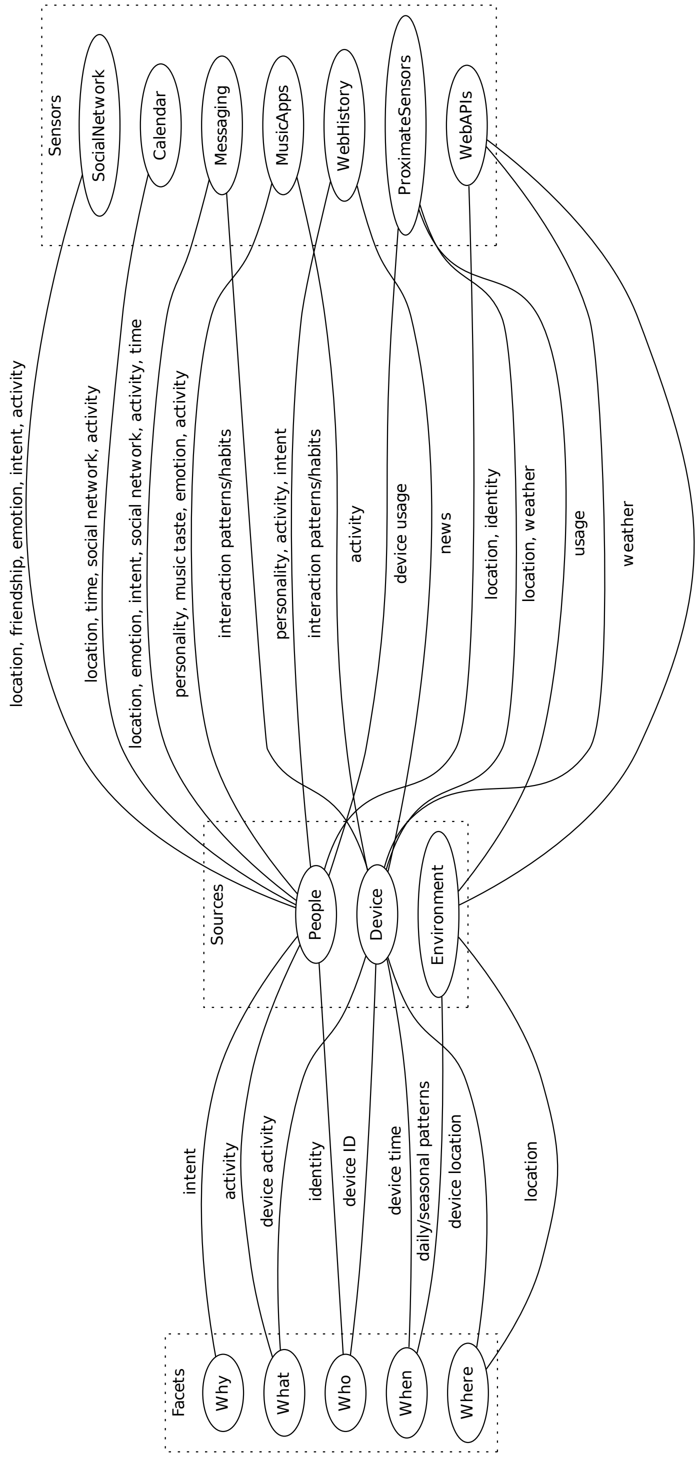


Figure 3.3: *Examples of virtual context sensors, with associated sources and facets. Edge labels from source to sensor show example instances, i.e. the relationship should be read as "<sensor> senses <edge> about <source>", e.g. "SocialNetwork senses friendship about people". Edge labels from facet to source should be read as "<source> provides data about <facet> through <edge>", e.g. "people provide data about 'who' through identity".*

the ubiquity of 3G (and the roll-out of 4G LTE) network coverage and cheap data, enables continual access to remote virtual sensors.

3.4 Context Data Fusion

Data fusion is the combination of two or more data into a compound datum, usually with the intent of increasing a particular property, e.g. accuracy, beyond what each individual datum can achieve. Data fusion has been used in applications such as target acquisition in defence systems [85] and complex database operations [28] in order to improve the quality of input data.

In context aware computing, data fusion has been applied to: smart home environments in order to improve the inference of occupant activity and location [95, 102], and office meeting detection using both a physical sensor array with probabilistic data fusion [169], and pressure sensors with logical data fusion [232]. It is a useful low level approach to improving the quality of context data before it reaches higher level inference processes.

3.5 Study: The Calendar as a Context Sensor

In this section, we perform an in depth study of one particular virtual context sensor: the calendar. The calendar is an interesting candidate for context sensing because – as Figure 3.3 shows – it can capture context data about people across all context facets. Moreover, modern calendars can be shared between groups of people for coordinating meetings and scheduling events. This raises the question of how useful a calendar-based sensor might be if we use data fusion to combine multiple users' calendars and other context together. As we will see in Chapter 4, inferring *meaning* (the "Why" facet) from context data can be a non-trivial problem, so if meaning – in this case calendar event names – is readily accessible through a virtual sensor, the effort of higher level inference could be reduced at the sensing layer.

The shared calendar has long been an effective tool for collaborative organisation and management, especially in office environments. Not only is it a useful indicator of people's presence and availability but many people use it for purposes such as archiving [178] and content management [79]. Shared calendar events can be a useful source of context as they contain data about groups of people that may otherwise be unavailable or unobtainable using other physical and virtual sensors.

Given the calendar's potential as a context sensor, how might we systematically analyse its sensing performance? The ubiquity of shared calendars in the workplace has resulted in computing infrastructure where network servers and databases store the shared calendar data of employees in a central location [170]. This network-based approach allows employees to view others' calendars and schedule meetings at available times without otherwise contacting the people involved. The office is therefore an ideal environment to study the calendar as a virtual context sensor, as: it is likely to be used often; office workers' location patterns are likely to be within a single building or site; and the calendar is usually available though some form of network.

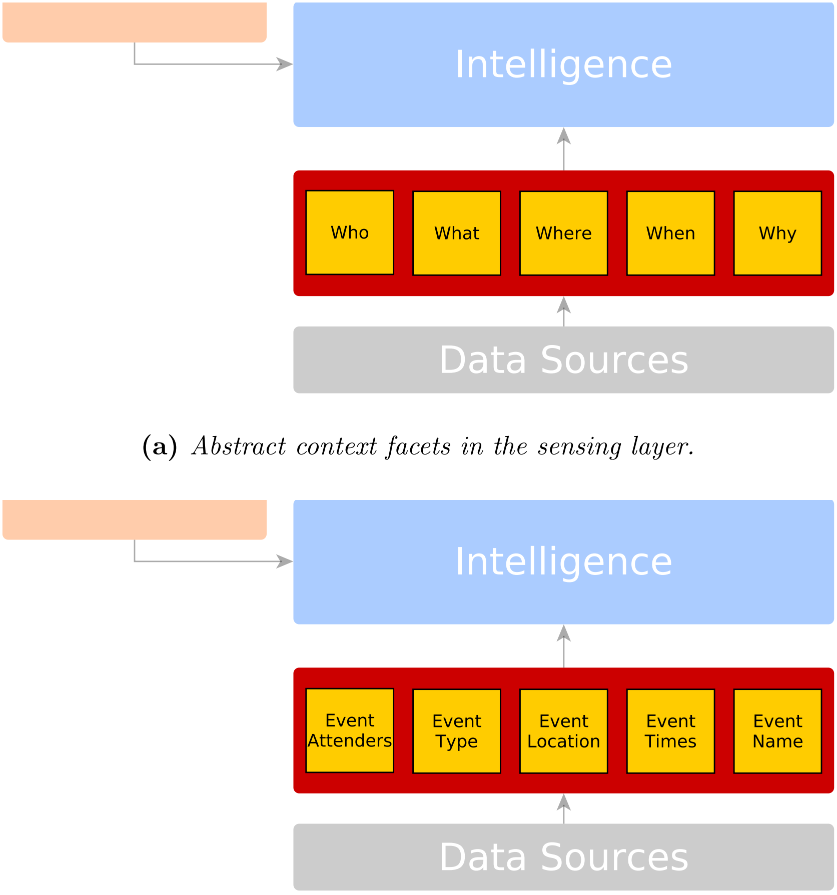
- We can distribute multiple context sensors within a single localised environment where potential participants spend most of their day.
- People are more inclined to share calendar data (and other context data) at work, where privacy may be less of an issue than in their personal lives [170].
- By using a relatively small physical environment such as an office means that we can employ both ethnographic and self-report methods to capture actual context. This will provide a basis for analysing the calendar's sensing performance.

We begin by outlining the key research questions for the study.

3.5.1 Research Questions

Before we begin outlining our approach, we revisit our design questions from Section 3.1 in order to establish a starting point for our study:

- Which facets of context do we need? To fully evaluate the calendar's potential as a context sensor, we would like to consider all facets in the Five Ws model; the calendar is certainly capable of sensing data about all five. Following Figure 3.4, these data are: event attenders ("Who"); event type, or category ("What"); event location ("Where"); event start and end times ("When"); and the event name, or subject ("Why").
- From what sources can we usefully sense context? We are mainly interested in people – particularly groups of people – who use and share calendar data on a daily basis.
- What sensors are available to us? Clearly the calendar is the key sensor, but we could also deploy further sensors, e.g. location sensors, in order to capture additional context data from our calendar users.
- Are there any constraints? The initial constraint is the environment for study. We have already noted the benefits of using an office environment but, in doing this,



(a) *Abstract context facets in the sensing layer.*

(b) *Concrete context facets for context sensing with the calendar.*

Figure 3.4: *Context facets that could be sensed with the calendar.*

we are restricting the generality of our potential findings, i.e. they may not readily generalise to calendar environments outside an office. Further constraints lie in user privacy (calendars could contain sensitive data that require secure capture and storage); office building infrastructure, safety and security constraints, e.g. company permissions to capture and store data; and availability of further sensors for deployment.

These questions provide us with a set of guidelines for approaching the study in a practical sense, but we must also address the concrete research problems and questions. The calendar alone is limited as a virtual sensor for a number of reasons. Firstly, it is unlikely to be a consistently accurate representation of the real world due to events not occurring, or its common use as a to-do list, i.e. users may use the calendar for purposes which the designers did not intend. Also, events may occur outside their allotted time window, and actual event attenders may differ from those invited. If a system were to use the calendar as a virtual sensor, it would ideally require as little deviation from the real events as possible. Data archiving or mining systems using the calendar for indexing could experience an impact on reliability for the same reasons. Secondly, the calendar does not provide dependable real time information. For example, within most enterprise instant messaging systems, e.g. Microsoft's Office Communicator, a user's availability is automatically changed to 'in a meeting' when a calendar event occurs. If the user is planning to attend the meeting late, or has left the meeting early, it is not reflected in her online presence. Thirdly, reminders and to-do list items are also commonly registered as events on such systems and again the user's availability will be listed as 'in a meeting' when in reality she is not.

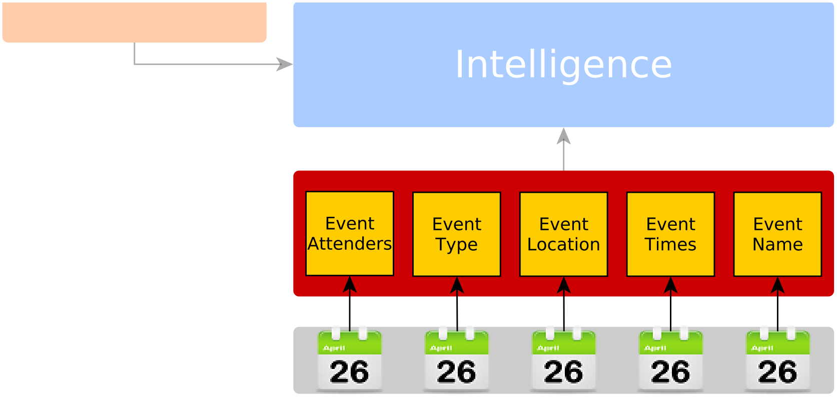
There are two key research questions that we aim to address with this study:

- **RQ 1.1**: How does the calendar perform as a virtual context sensor? This is the key question; how good is the calendar at sensing context? If we can measure its sensing performance, we can provide a quantifiable estimate of its use as a standalone context sensor.
- **RQ 2.1**: To what extent does combining the calendar with other sources of context data affect overall sensing performance? This relates to data fusion. How might we fuse other context data with the calendar, and does data fusion improve sensing performance beyond the calendar alone?

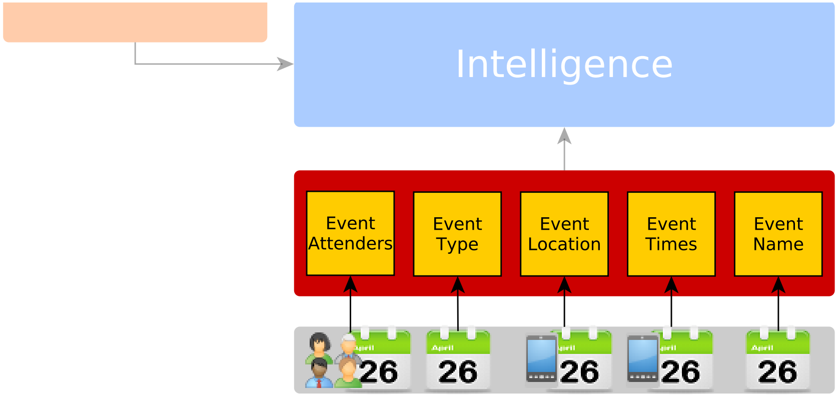
The aim of the study, as illustrated in Figure 3.5 is to address these questions by: (i) comparing the calendar with actual events; and (ii) combining the calendar with other context sensors and comparing the combined output with actual events.

The following sections outline directly related research, our study approach and our results. As we shall see, the results show that genuine calendar events, i.e. events on the calendar that actually happened, make up only a small fraction of the total calendar events, resulting in a low precision measure; suggesting that the calendar alone is a low fidelity context sensor in practice. Moreover, there are a certain number of events that do occur but are not on the calendar – so called *ad hoc* events – which affect the calendar's recall measure. Furthermore, for the events that do appear on the calendar, we show that there are significant deviations in event times when compared with their real world counterparts, as well as poor location specification in calendar entries. Interestingly, one result shows that event attender lists are very representative of reality for the calendar events that *do* occur.

We approach our study in two parts according to RQ 1.1 and RQ 2.1. Firstly, we measure the performance of the calendar as a standalone virtual context sensor, and secondly, we present two simple algorithms for logical data fusion that combine the calendar with location data from mobile devices and social network data from email



(a) *RQ 1.1 addresses context sensing with the calendar alone.*



(b) *RQ 2.1 addresses the fusion of other context sensors with the calendar.*

Figure 3.5: *Visualising the research questions.*

networks to sense genuine events in real time. We apply these algorithms to the data gathered in the field study, showing that event sensing can be significantly improved through data fusion of the calendar (a virtual sensor), an email network (a virtual sensor) and Bluetooth (a physical sensor) location data. Consequently, useful context data within the calendar can be uncovered, enabling the development of new applications or improvements to existing applications that make use of people's presence and availability.

3.5.2 Related Work

Before describing the study approach, we contrast our work with directly relevant literature. Employing mobile devices within context aware systems and applications are active research areas. Early work by Schmidt *et al.* [207] used physical and logical sensors on a mobile device to demonstrate situational awareness with a similar layered model approach to ours, and Indulska and Sutton [103] discuss the idea of physical, virtual and logical sensors when applied to location management in pervasive systems. Of particular relevance here is the authors' layered framework, which features a Fusion layer combining abstracted outputs from the different sensor types to enhance location information given by each of the standalone sensors. Fogarty *et al.* [72] present a context aware communication client that uses data fusion to provide a better interpretation of how interruptible a user may appear to their colleagues.

Forecasting activity, presence and availability through the use of calendars are also popular research topics. Various projects have investigated the usefulness of the calendar in coordination and collaboration [162]; how the calendar is used [178] and applications to augment the shared personal calendar [227]. Mynatt and Tullio have contributed a number of studies on the use of the calendar and its applicability to future availability prediction. In [160] they present a calendar application as a sensor that provides a likelihood of users' future presence and location. The application, Ambush, uses a Bayesian model to predict attendance likelihood for calendar events based on previous attendance. In this work, they also show that co-workers in enterprises use their shared calendar to 'ambush' colleagues for *ad hoc* meetings when they are not busy. In [228] they discuss the deployment of the application and implications of using forecasting in groupware system design.

In [226], perhaps the most relevant to our work, Tullio states that during his studies, events were attended between 52% and 63% of the time. Citing an unpublished study by Bradner, he notes that calendars are often cluttered with events that were not attended, highlighting his desire to provide a more informed interpretation of users' schedules. Further work on the calendar's use in the workplace has been undertaken by Palen and Grudin [170]. They show that office workers frequently use shared calendars to infer the presence and availability of their colleagues.

Horvitz *et al.* [97] constructed an interesting system that uses users' desktop PC activity to build a Bayesian probabilistic model of presence and availability. By analysing events such as mouse movement coupled with application use, the prototype application attempts to estimate when users are likely to return to their desks, or whether a particular time of day is preferable for a meeting given potential attendees. However, there is no evaluation of system performance, so it is difficult to compare how effective the authors' approach is at capturing context in real time using the calendar.

Research has also been conducted on the concept and definition of events, as well as their identification through various sensory inputs. Westermann and Jain [238] present a common model to describe the facets of an event, broadly classified around key areas of context, i.e. temporal, spatial and social. Event detection is discussed at length by Xie *et al.* [242], who investigate and classify various event detection systems and their uses in problems such as media management and data mining. They draw on a 5W1H model of event classification which is similar to the model presented by Westermann and Jain. They also look at the role of context when detecting events, alluding to the advantage of *a priori* knowledge, or planning, in event classification. The event detection analysed by this work is generally undertaken *post hoc*, i.e. mining post-event multimedia in the attempt to detect the event itself. In contrast, our work focuses on the real time aspects of event detection, detecting the events as and when they happen.

Eagle and Pentland's BlueAware and BlueDar [65] systems – used in the collection of the popular Reality Mining data [67] – use similar methods to ours when identifying co-present system users. They fuse user profile data with this co-presence information in an attempt to induce 'social serendipity' between proximate users who do not know each other. Real time meeting detection is also investigated by Wang *et al.* [232], who present a meeting identification system that uses data fusion. They attempt to measure meeting start and end times through pressure sensors in seats, with a 95% success rate. Other research into the importance of meeting semantics, knowledge of meetings and capture of meeting metadata is discussed in [84]. This work suggests that there is value in the consistent and accurate semantic capture of meetings and the advantages these data bring to knowledge management and legacy searching problems.

3.5.3 Approach

Our field study ran for 6 weeks in an office building for a business group (approx. 200 employees) within a large telecommunications company (approx. 85,000 employees). The main employee roles within this business focus on software development and engineering. Scheduled meetings are commonplace among the employees, and the environment is representative of many modern open plan offices. The company uses the Microsoft Outlook application with Exchange Server as a shared calendar tool.

We recruited 20 participants from within two closely related teams, with 11 (3 female) and 9 (1 female) participants from each team respectively. One participant in each team had a managerial role while the remainder were software developers working under the managers. We chose the participants from these two teams due to their frequent intra and inter team meetings in particular meeting areas. All were familiar with mobile devices and had some experience with the Windows Mobile operating system. 20 participants were chosen as they formed two complete teams and provided a sample size that would allow us to reliably monitor them ethnographically whilst generating enough data for statistical analysis.

The following sections describe the data collected during the study, and the methods used to undertake the collection.

Capturing Calendar Events

In order to capture our participants' calendar events, we obtained programmatic access to their Outlook application via the company's Exchange Server; we had access to these throughout the duration of the study. Calendar events were captured 'live', i.e. we recorded the entries in real time, storing any changes made by the participants during the study, such as amended invite lists, times, locations and event names. The Exchange Server that manages the Outlook calendar events assigns each event object a unique ID, so every event had a single identifier even if it were stored in multiple calendars. Events such as private appointments that were not accessible through the shared system were ignored.

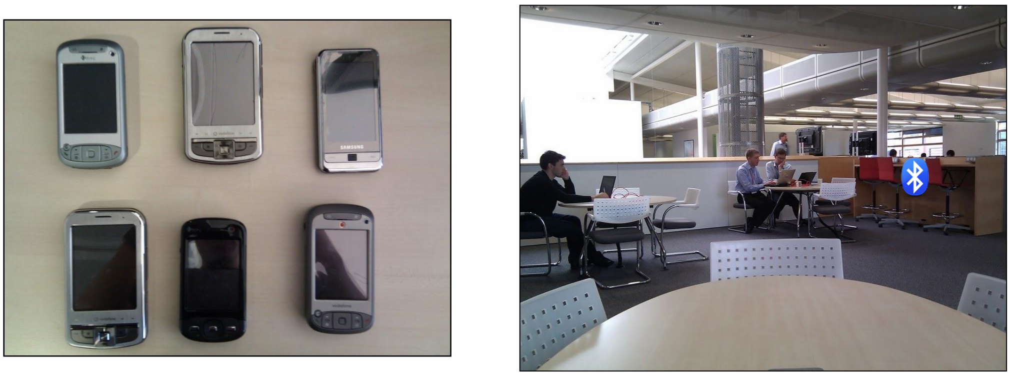
Capturing Actual Events

Actual events represent what actually happened in terms of meetings involving two or more of our participants. Our record of these was obtained through a combination of three methods: ethnographic observation, participant interview and participant self-report. We restricted our study to four primary meeting areas for the two teams – all within a few seconds' walk from each other (see Figure 3.7). For ethnographic observation, we set up a temporary workspace within the team environment and observed the participants during their working days; recording any events involving two or more of our participants that took place in the four meeting areas. We could not monitor all the participants all of the time however, so we instructed them to keep an event diary for the 6 week period, within which they recorded details of their workplace meetings (see the diary template in Appendix A.1). Finally, we conducted weekly interviews with participants. This included examining their diaries for the week, validating our observed data – i.e. the events we had recorded – against their diary entries and verifying our recorded events with them through questioning and discussion.

Capturing Context Data

We collected the following additional context data about our participants throughout the study:

- **Location**: Each participant was given a mobile device running the Windows Mobile operating system. (The range of device types is shown in Figure 3.6a.) We built a small application to run on the devices that performed a Bluetooth scan of the surrounding environment at 2 minute intervals. After each scan had completed, the application uploaded the timestamped results to our server using



(a) *The range of devices provided to the participants for the duration of the study. Each runs Windows Mobile 6.1 and is Bluetooth compatible.*

(b) *Photograph of the upper-right meeting area in Figure 3.7. The Bluetooth icon shows where the static scanning device was placed – hidden in the wooden cabinet row.*

Figure 3.6: *Photographs of the study devices and office locations.*

either 802.11 WiFi or a GPRS mobile data connection depending on connectivity, i.e. if one connection failed, the device would switch to the other. In order to estimate participant location, we placed 4 static devices in known positions within the workplace. These locations, shown in Figures 3.7 and 3.6b, were the key meeting areas for the two teams, and the static devices served as identifiers for each location. Each device performed a Bluetooth scan at 1 minute intervals and uploaded the timestamped results to our server. Thus, if a participant were to move within the 'hotspots' in Figure 3.7 there would be a good chance, subject to the usual vagaries of Bluetooth scanning [168], of their mobile device reporting a Bluetooth sighting of a static device and *vice versa*. The area in which the team desks are located was not covered by a static device. This was to minimise the risk of the static devices interfering with each other or reporting ambiguous results due to participants being sighted in two hotspots at once. Although ambiguity was addressed, this decision did affect the sensing latency of participants' event exit times.

- **Contacts**: In addition to accessing participants' calendars, we also captured their manually created contacts, i.e. non-corporate address book contacts, of each participant through their Microsoft Outlook application. These too were recorded 'live'; i.e. when contacts were added, changed or deleted the action was communicated to our server. Existing contacts, i.e. contacts added to the address book before the study, were also captured when the applications were installed on the participants' computers.

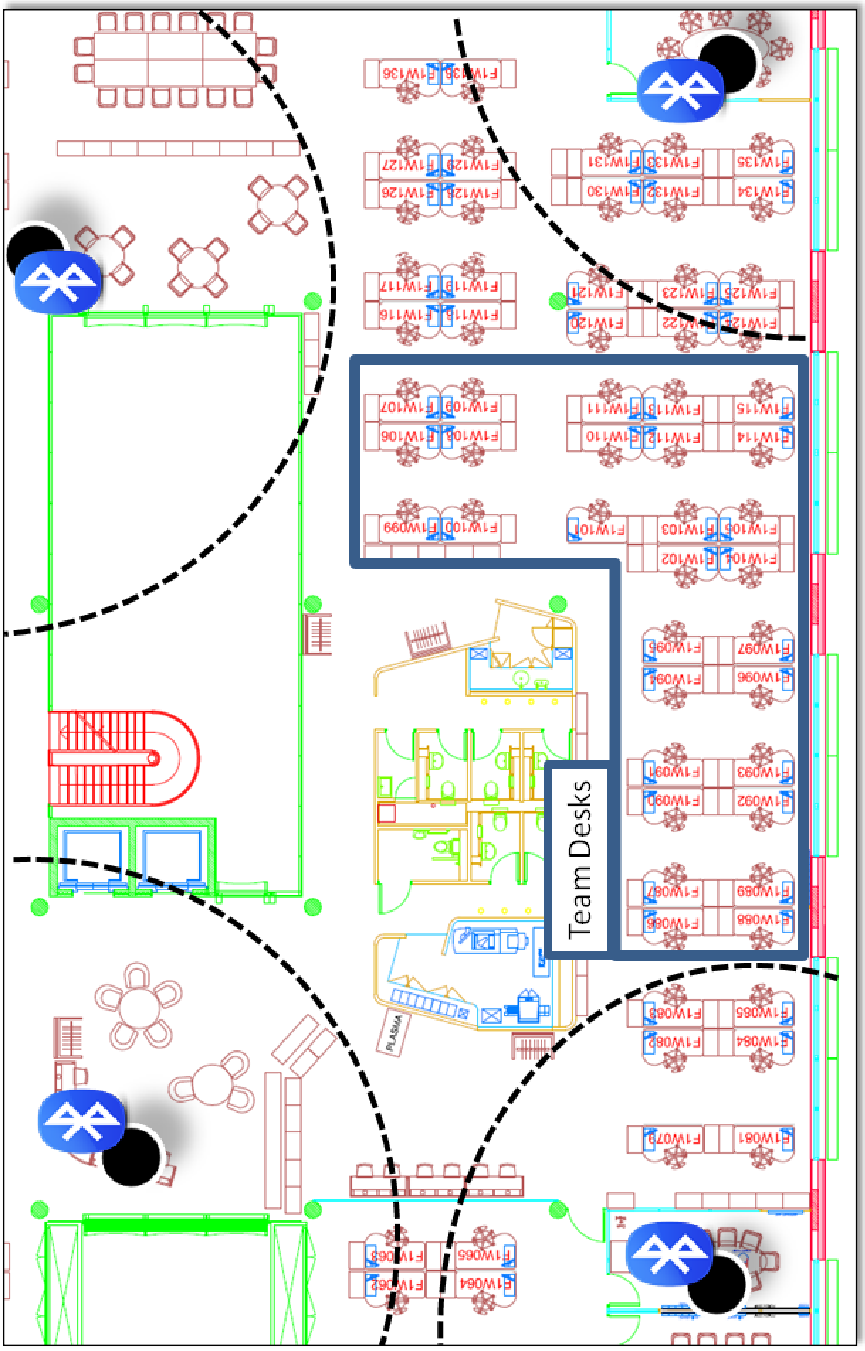


Figure 3.7: *Floor plan showing the office environment for the study, including the desk area of the two teams. The 'hotspots' indicate the placement of the static devices, with approximate ranges shown. The lower two devices (when orienting the image to landscape) were placed in meeting rooms, whereas the upper two were in open meeting areas.*

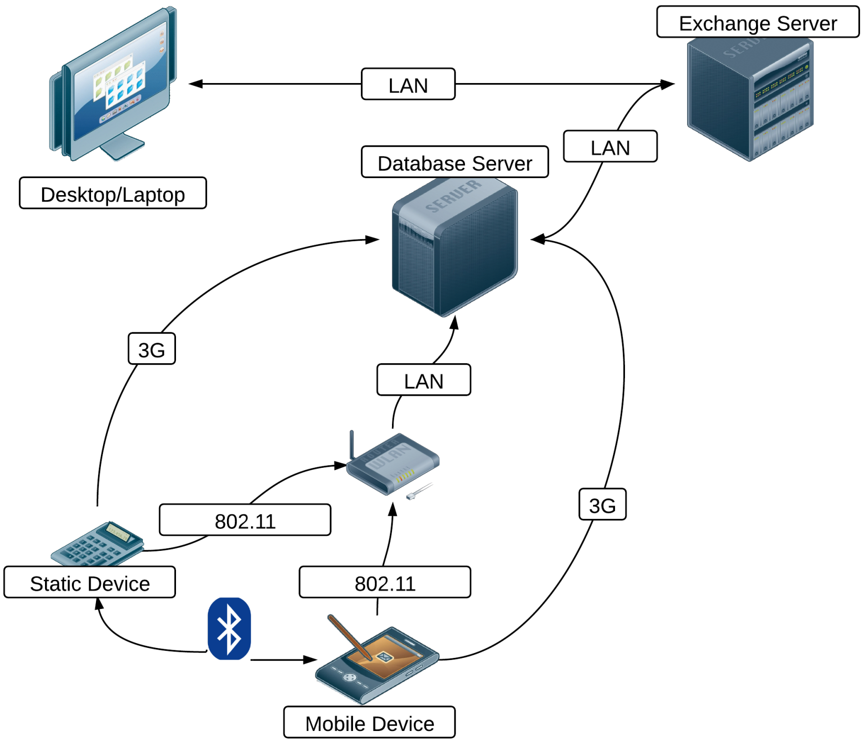


Figure 3.8: *The network architecture for the office study, showing mobile and static Bluetooth nodes connected to our database server; either by 3G or 802.11 WiFi. Calendar and contacts were supplied by the office's Exchange Server, to which participant desktop/laptop machines connected through the Microsoft Outlook application on their computers.*

Figure 3.8 shows the network architecture of the field deployment, with the mobile devices, servers and network connections between them.

Categorising Calendar Events

In order to measure the calendar's accuracy as a virtual context sensor, we need to define what is 'signal', i.e. useful context data, and what is 'noise', i.e. irrelevant data. In order to do this, we categorise the calendar events into sets according to various observed characteristics:

- **Genuine Event** (G): A shared online calendar event involving one or more study participants that maps to an actual event.
- **Placeholder Event** (P): A shared online calendar event involving one or more study participants that does not map to a actual event because no actual event occurred, e.g. a recurring daily meeting that does not occur on a particular day.
- **Personal Reminder** (R): An online calendar 'event' created by a participant

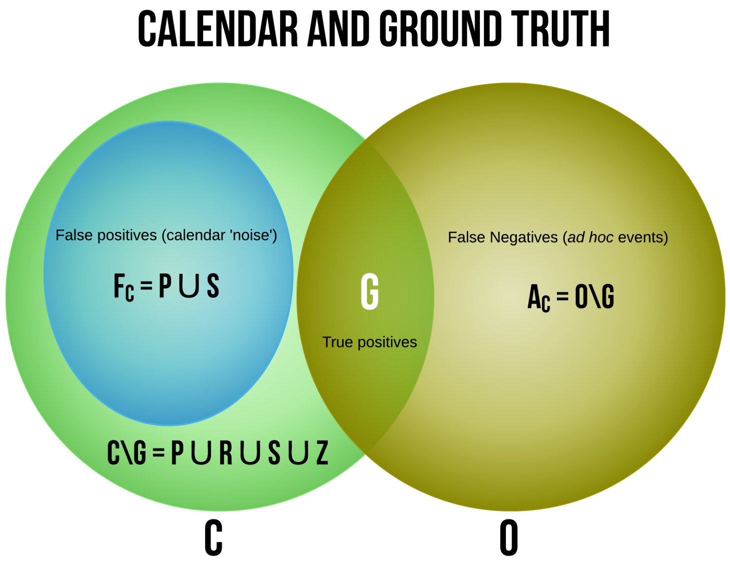

CALENDAR AND GROUND TRUTH
False positives (calendar 'noise')
Fc = P U S
G
True positives
False Negatives (ad hoc events)
Ac = O\G
C\G = P U R U S U Z
C
O

Figure 3.9: *Measuring performance: the set C contains all calendar events for our participants. O contains the actual events. C is the union of calendar event sets as described in Section 3.5.3. The set $G = C \cap O$ contains true positives, i.e. calendar events that occurred; $C \backslash G = P \cup R \cup S \cup Z$ contains all calendar events that did not occur – for our analysis we remove R and Z by simple elimination rules to leave the set of false positives $F_C = P \cup S$, i.e. calendar 'noise'; $A_C = O \backslash G$ are false negatives, i.e.* ad hoc *events that occurred but did not appear in the calendar.*

simply as a reminder to herself, e.g. 'Backup Files', without inviting anyone else.

- **Shared Reminder** (S): A shared online calendar event created as a reminder to two or more study participants, with 'attenders' 'invited' only to enable the sharing.

- **Out of Scope** (Z): A shared online calendar event that: (i) involves a single study participant and other people not involved in the study or; (ii) was external to our meeting areas e.g. at a different site or; (iii) is outside office hours.

These sets are disjoint, and the entire calendar set (C) is the union of these category sets:

$$C = G \cup P \cup R \cup S \cup Z \tag{3.1}$$

Performance Measures

To measure performance, we use the set O as the set of observed calendar events, i.e. actual events. Figure 3.9 visualises the sets O and C, detailing our performance metrics. These metrics are detailed as follows:

- True positives: the set of genuine calendar events, $G = C \cap O$.
- False positives: $F_C = P \cup S$. Once Z is removed from C, a simple exclusion rule can be applied to distinguish the personal reminders R from the other categories: ignore all events with fewer than two invited attenders (including the calendar event creator). However, it is not so trivial to differentiate between a genuine event G, a placeholder event P and a shared reminder S as they are all in exactly the same format in the online calendar and all have two or more invited attenders.
- False negatives: $A_C = O \backslash G$. These are *ad hoc* events that occurred but did not appear in the calendar.

Using these metrics, we can measure the calendar's precision p:

$$p = \frac{|G|}{|G| + |F_C|} \tag{3.2}$$

Where $|G|$ represents the cardinality (size) of set G. Furthermore, we can measure the calendar's recall r:

$$r = \frac{|G|}{|G| + |A_C|} \tag{3.3}$$

Thus, for our overall performance measure, we use the harmonic mean of precision and recall – the F1 score:

$$\text{F1} = 2\frac{pr}{p + r} \tag{3.4}$$

We also measure the location sensing performance for the events in G using the F1 score in Equation 3.4 with the following metrics:

- True positives (location): calendar events in G with a correct location description.
- False positives (location): calendar events in G with an incorrect location description.
- False negatives (location): calendar events in G without a location description.

Category	Symbol	Contribution
Genuine Event	G	38 (0.08)
Placeholder Event	P	152 (0.32)
Personal Reminder	R	232 (0.49)
Shared Reminder	S	52 (0.11)
Out of Scope	Z	120 (n/a)

Table 3.1: *The complete set of calendar events C and the measure of contribution for each subset to C.*

To measure the per-event attender classification performance for the events in G, the metrics are defined as follows:

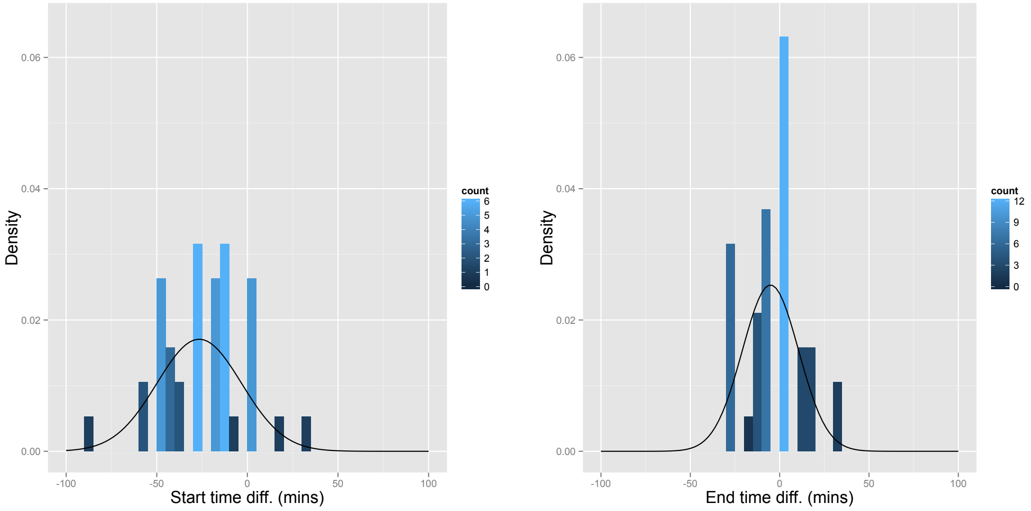
- True positives (per-event attenders): The number of attenders who are both on the event's invitation list and attended the event.
- False positives (per-event attenders): The number of attenders who are on the event's invitation list but did not attend the event.
- False negatives (per-event attenders): The number of attenders who are not on the event's invitation list but did attend the event.

We can measure the per-event precision, recall and F1 score for the attenders, and we can aggregate them to give an average performance measure.

3.5.4 Results: Calendar Performance

By the end of the field study, we had collected 594 unique online calendar events from the participants. In contrast, we recorded only 38 distinct real-world events involving two or more participants, each of which corresponded to one of these calendar events. In Table 3.1, we list the number of calendar events according to the categories defined in Section 3.5.3. Events in set Z are beyond the scope of our analysis since we are studying only a subset of employees from the whole business, in a sample location and during normal working hours. Excluding Z from the set of 594 calendar events leaves 474 in scope events for us to consider. Table 3.1 also lists the proportion of each category, excluding the events in Z. In addition to these events, we also observed 6 *ad hoc* events.

Figure 3.10 shows histograms of calendar event start and end time differences in G – to the nearest 5 minutes – relative to the equivalent observed event times, i.e. $t = 0$ represents a actual event occurring within 5 minutes of its calendar entry; $t < 0$ represents the actual event occurring *after* its calendar entry; and $t > 0$ represents the actual event occurring *before* its calendar entry.



(a) *Start time differences.*

(b) *End time differences.*

Figure 3.10: *Histograms showing time difference densities of calendar events in G (with fitted maximum likelihood normal density functions) relative to their equivalent observed events. A negative time therefore indicates a delay in actual event time relative to calendar event counterparts.*

The start times of real events are significantly later than the calendar indicates ($t_{37} = -7.01; p < 0.01$; Student's t-test), though end times are not significantly different ($t_{37} = -2.01; p = 0.051$; Student's t-test). Table 3.2 lists the summary statistics for the calendar's performance as a context sensor, including location and attender performance measures.

Statistic	**Value**
Event precision	0.16
Event recall	0.86
Event F1	0.27
Start time $(\bar{x}_s, s_s)$	$(-26.6, 23.4)$
End time $(\bar{x}_e, s_e)$	$(-5.1, 15.7)$
Location $\{p, r, \text{F1}\}$	$\{1, 0.11, 0.2\}$
Mean attender precision	0.93
Mean attender recall	0.94
Mean attender F1	0.92

Table 3.2: *Summary statistics for the calendar's performance as a virtual sensor of context. Variables are: $\bar{x}$ = sample mean; s = sample standard deviation; p = precision and r = recall. Times are in minutes.*

3.5.5 Discussion

In this section, we discuss our initial findings and address RQ 1.1. As Table 3.1 shows, nearly half the events in the study are actually personal reminders R. The set of placeholders P accounts for a third and the set of genuine meetings G accounts for only 8%, outweighed even by the set of shared reminders S.

RQ 1.1: How does the calendar perform as a virtual context sensor?

The overall F1 score of 0.27 suggests that the calendar alone is an imprecise and 'noisy' virtual context sensor. This is chiefly due to the low precision measure, i.e. a high number of false positive events F_C. Along with the 38 genuine events in G, the calendar contains 204 false events; which are composed of shared reminders and placeholder events. A recall score of 0.86 shows that the majority of events in the workplace do appear on the calendar in some form, i.e. *ad hoc* events are in the minority. However, the F1 score is dominated by the 'noise' of the false positives.

The time differences between the calendar and actual events in G illustrate the cause of availability discrepancies described earlier, where a presence and availability application such as Microsoft Communicator will list a user's presence as 'in a meeting' when in reality she is not. As the results show (see Figure 3.10 and also the calendar condition in Figure 3.14b), the majority of actual events in this particular office start later than indicated by the calendar, and the sample standard deviation figures show a large variability between calendar and observed event start and end times.

Why was this? One of the main observed causes of start time variance was participants simply turning up to meetings late, though a reasonable number of meetings were reorganised at the last minute without updating the calendar entry. One meeting started $\approx$ 90 minutes later than scheduled (Figure 3.10a) as it was a 'block booking' for a meeting room.

The low location F1 score of 0.2 is a result of the calendar event location field not being consistently populated by participants, suggesting that the calendar is not a good sensor of event location. Although participants occasionally used the location field (and correctly, hence the precision of 1), it was usually empty, resulting in a low recall score of 0.11.

Interestingly, the calendar *does* appear to be a good sensor of event attendance. With the high F1 score of 0.92, it appears that the majority of participants attended events that they were invited to, with only a few absentees or additional (unlisted) attenders. This has interesting implications for practical application, particularly for presence and availability applications: if we could extract the events in G from the calendar events C, we could increase our belief in users' reported presence and availability. The problem with the standalone calendar, of course, is that G is indistinguishable from P and S without knowledge of further context.

To summarise, we found that in a typical office calendar, the vast majority of 'events' are reminders or placeholders, and few were actually representations of genuine real-world events. We also found that the similarity between actual events and their calendar equivalents is variable, and that the calendar is not a reliable sensor of event times or locations (see Table 3.2 and Figure 3.14). It is, however, a good sensor of event attendance; but this information is hidden among the false events and not easily discernible without *a posteriori* knowledge and additional context. Thus, without additional exogenous knowledge of context, it is difficult to differentiate between genuine events, placeholders and shared reminders, making the calendar alone an unreliable virtual sensor.

3.5.6 High-level Implications and Limitations

Here we discuss the implications of our work in relation to the high-level research question: RQ 1. RQ 1 asks what entities might we consider as virtual context sensors, and RQ 1.1 asks how the calendar performs as a virtual context sensor. The key implication for the former in answering the latter is that the calendar *can* be considered as a virtual context sensor, but it is *not* a good quality one. We have shown that the calendar contains multiple, useful, context data relating in some part to the majority of the Five Ws context facets. Even though it is 'noisy', it is still a veritable source of context data and, as such, should be considered a viable, canonical virtual context sensor.

The key limitation in studying one virtual sensor in such detail, however, is that the breadth of the work is lacking. That is, without repeating similar studies for other virtual sensors, we cannot fully answer RQ 1. These immediately sets an agenda for further research in order to explore RQ 1 in greater detail: given our approach in studying the calendar, use the same methodology to study other potential sensors such as email, social media content and music players. An ideal output might be a database or record of virtual sensors – similar to those in Figure 3.3 – in which their usefulness as a virtual context sensor may be catalogued for researchers and software developers to use in their work.

3.5.7 Data Fusion: Combining Other Context Data

Here we outline our context data fusion approach and the results observed when comparing the combined data output with actual events and the calendar.

Data Fusion Algorithms

Given the three context sensors in our study: the calendar, location and social network, how can we begin to fuse these data and classify the events in real time? We designed two candidate data fusion algorithms that classify events according to relationships between users' context, e.g. shared locations, social connections or shared calendar events. The algorithms output events by partitioning users into groups according to context graph ties at a given moment in time, and annotating the groups with calendar data. But where should we start? What should we use as the first indicator that an event might be occurring?

Let us consider using social network data to detect event occurrence: searching for connected components in a global social network would require knowledge of a possibly vast and temporally dynamic social network in which changes to the network over time, e.g. creation of new edges, vertices or clusters, could somehow be related to event occurrence. This approach is unlikely to provide a consistent indication of event occurrence, however, so we dismiss initially partitioning users through their social network data. What about using the shared calendar events to trigger data fusion? In this case the algorithm would be triggered at the start time of each calendar event. As we have seen, however, the calendar alone is not a good indicator of context – the algorithm would have to be robust to false positive events, and it would miss *ad hoc* events.

This leaves location or, more specifically, co-location. Could a sudden gathering of people indicate the start of an event? This seems likely, but how could we integrate the calendar and social network data into the process? Our two candidate algorithms integrate these data in different ways, and we will compare their performance on the study dataset in the next section. They are formally described in Algorithm 1 and Algorithm 2 respectively, and example visualisations are shown in Figure 3.11. The algorithm descriptions are as follows:

1. **Algorithm 1** (Figure 3.11a). At the time of execution, users are connected according to co-location. They are then split into subcomponents according to calendar event 'ties', i.e. shared calendar events that are listed as ongoing at the time of execution. The remaining co-located users are added to the sets according to their social ties, i.e. each is assigned to the set with the majority of social ties to themselves (equal sets are broken at random). Those without any ties to any

Algorithm 1 Co-located people are connected through calendar graph ties. Those remaining are connected to these components through social network ties; and events are named using the calendar event of each connected component.

```
Input: a set $P$ of people.
Output: a set $V$ of events that are currently occurring.
$V \leftarrow []$                                        ▷ Initialise events data structure.
$U \leftarrow$ LocationConnectedComponents($P$)                  ▷ Location search.
for all $u \in U$ do
    if $|u| < 2$ then
        continue
    end if
    $C \leftarrow$ CalendarConnectedComponents($u$)              ▷ Calendar search.
    $R \leftarrow$ Complement($u$, Union($C$))          ▷ Those without calendar entries.
    $A \leftarrow []$                              ▷ Initialise empty ad hoc data structure.
    for all $r \in R$ do
        AddToSocialMajority($r$, $C$)           ▷ Add $r$ to group with most social ties.
        if $\neg$Connected($r$) then
            Append($r$, $A$)                             ▷ No ties, ad hoc candidate.
        end if
    end for
    $D \leftarrow$ SocialNetworkConnectedComponents($A$) ▷ Connecting ad hoc candidates
    Append($D$, $C$)                           ▷ Append ad hoc connected components.
    for all $c \in C$ do
        if $|c| \geq 2$ then
            Append(CreateEvent($c$), $V$)  ▷ Event has name, location and attenders.
        end if
    end for
end for
return $V$
```

set are connected to each other through their social network – these connected components form an *ad hoc* event.

2. **Algorithm 2** (Figure 3.11b). At the time of execution, users are again connected according to co-location. They are then split into connected subcomponents according to social network ties, i.e. each subcomponent is a subgraph within the co-located social network. Finally, each subcomponent is labelled with the ID of the calendar event listed in the majority of the subcomponent's users' calendars (ties are broken randomly); subcomponents are then connected if they share an event ID. If there are no calendar events in any of calendars, the event is classed as *ad hoc*.

Figures 3.11b and 3.11a show how – at a given timestep k – these algorithms can result in different event outputs given the same input data.

Algorithm 2 Co-located people are connected through social graph ties. Calendar entries are used to name these components and connect events.

```
Input: set set $P$ of people.
Output: a set $V$ of events that are currently ongoing.
$V \leftarrow []$                                        ▷ Initialise events data structure.
$U \leftarrow$ LocationConnectedComponents($P$)
for all $u \in U$ do
    if $|u| < 2$ then
        continue
    end if
    $S \leftarrow$ SocialNetworkConnectedComponents($u$)        ▷ Social network search.
    $E \leftarrow []$                                    ▷ Initialise empty event data structure.
    for all $s \in S$ do
        Append(CreateEvent($s$), $E$)                    ▷ "Majority wins" event name policy.
    end for
    $C \leftarrow$ CalendarConnectedComponents($E$)             ▷ Connected events merged.
    for all $c \in C$ do
        if $|c| \geq 2$ then
            Append($c$, $V$)
        end if
    end for
end for
return $V$
```

To classify events in real time, Algorithm 3 executes periodically to update event start and end times. It continually maintains two sets of events, E_k and E_{k-1} for the current and previous timesteps respectively. A new event is started if it is an element of E_k but not E_{k-1}, and ended if it is an element of E_{k-1} but not E_k.

Performance Measure

Each data fusion algorithm in Section 3.5.7 outputs a set of classified events D. Figure 3.12 illustrates our comparison between the classified events in D and the observed events O.

To measure the performance of the algorithms, we again use the F1 score for classification accuracy (see Equation 3.4), defined using true positives (T), false positives (F_D) and false negatives (A_D). These we define as follows, following Figure 3.12:

- True positives: $T = D \cap O$; which are events in D that map to a actual event in O. We assigned the actual events with calendar entries the same unique ID as their corresponding calendar entries, so successful event identification is measured by comparing the ID of the classified event with the ID of the actual event. *Ad hoc*

Algorithm 3 Classifying events in real time.

```
$E_{k-1} \leftarrow []$                    ▷ Initialise previous timestep empty events data structure.
$E_k \leftarrow []$                        ▷ Initialise current timestep empty events data structure.
while True do
    $E_{k-1} \leftarrow E_k$                ▷ Previous events' pointer updated.
    $E_k \leftarrow$ ExecuteDataFusionAlgorithm()    ▷ Algorithm 1 or 2.
    for all $e_k \in E_k$ do
        if $e_k \notin E_{k-1}$ then
            StartEvent($e_k$)              ▷ Write start time.
        end if
    end for
    for all $e_{k-1} \in E_{k-1}$ do
        if $e_{k-1} \notin E_k$ then
            EndEvent($e_{k-1}$)            ▷ Write end time.
        end if
    end for
    Wait($t$)                              ▷ Timestep period.
end while
```

events are manually identified. In addition, the classified event must occur at the same time as the actual event. 'At the same time' means that the time window of the classified event overlaps the time window of the actual event.

- False positives: $F_D = D \backslash T$ are events in D that do not map to any event in O. That is, an event whose ID either: (i) does not match the ID of any actual event or; (ii) does match the ID of a actual event but does not overlap the actual event in time.

- False negatives: $A_D = O \backslash T$ are events in O that do not map to any event in D. These include calendared events in O that are not classified, and *ad hoc* events that are not classified.

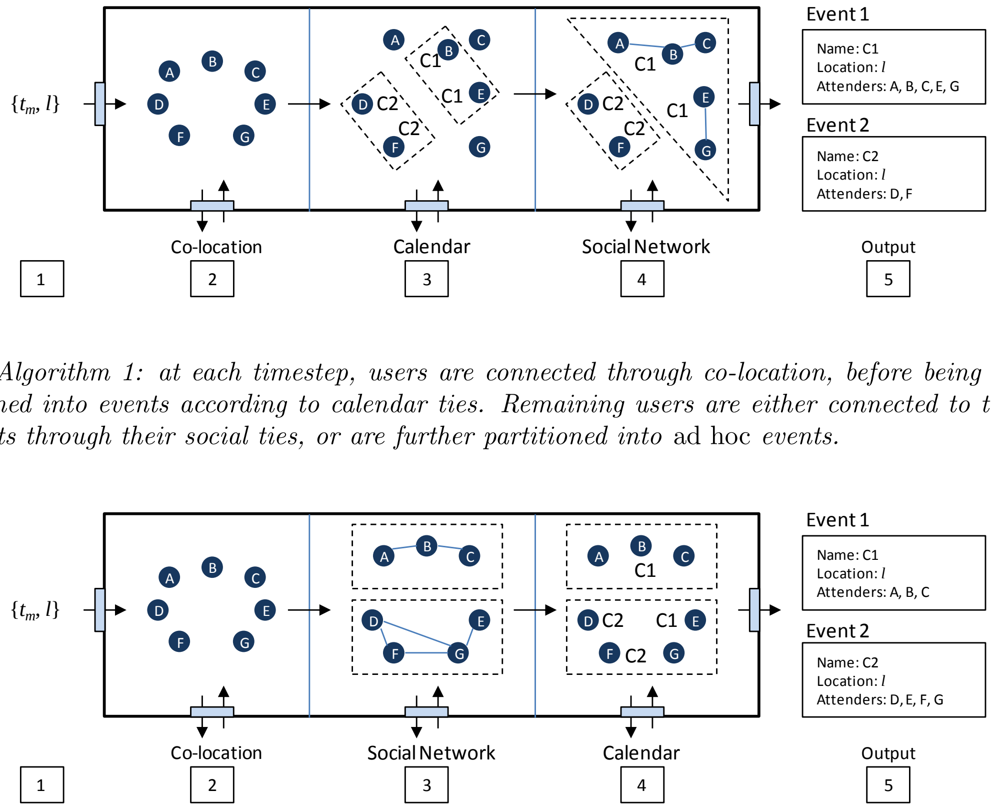
As with the calendar time measures, start and end time differences are measured relative to the equivalent observed event times; thus, a negative time difference indicates that a classified event 'started' or 'ended' before its observed counterpart. These are also rounded to the nearest 5 minutes.

As before, we measure location performance for classified events in T using precision, recall and F1 score; with the base metrics as follows:

- True positives (location): a classified event with a correct location.

- False positives (location): a classified event with an incorrect location.

- False negatives (location): a classified event with no location.



(a) *Algorithm 1: at each timestep, users are connected through co-location, before being partitioned into events according to calendar ties. Remaining users are either connected to these events through their social ties, or are further partitioned into* ad hoc *events.*

(b) *Algorithm 2: at each timestep, users are connected by co-location, then partitioned into events according to their social network. Events are classified according to a "majority wins" calender policy. If there are no calendar events in the partitioned group, then an* ad hoc *event is classified.*

Figure 3.11: *Visualising the data fusion algorithms on an example data set.*

As the algorithms always classify a location, location recall is always 1.

Again, to measure the per-event attender classification performance for events in T, we use the following metrics:

- True positives (per-event attenders): The number of attenders who are both on the classified event's attender list and attended the event.
- False positives (per-event attenders): The number of attenders who are on the classified event's attender list but did not attend the event.
- False negatives (per-event attenders): The number of attenders who are not on the classified event's attender list but did attend the event.

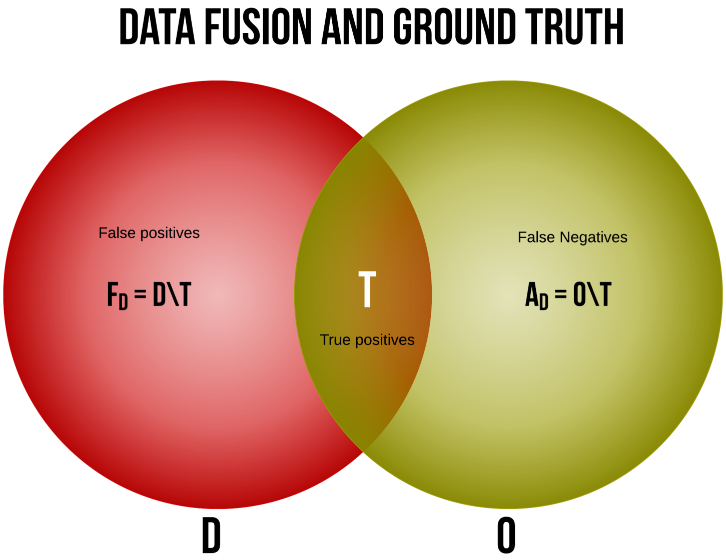


Figure 3.12: *Comparing the data fusion outputs with actual events*

3.5.8 Results

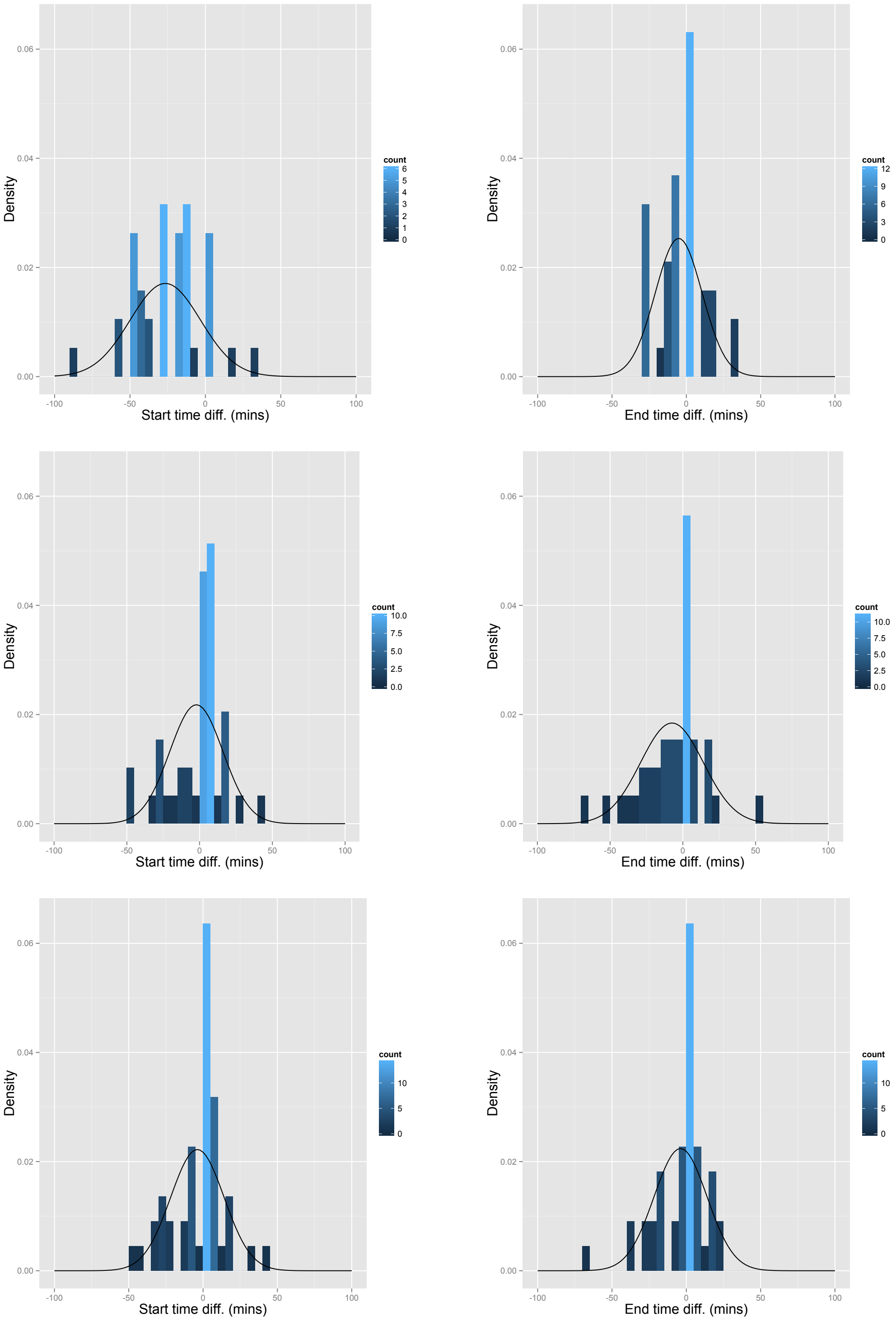
The context data collected during the study were input to the data fusion algorithms. Here we describe the results for each process when their outputs are compared with the observed event data.

For Algorithm 1, the output was $|T| = 38$, $|F_D| = 14$ and $|A_D| = 6$; thus $p = 0.73$ and $r = 0.84$. For Algorithm 2 the output was $|T| = 43$, $|F_D| = 32$ and $|A_D| = 1$; thus $p = 0.57$ and $r = 0.97$. These give F1 scores of 0.78 and 0.72 for Algorithms 1 and 2 respectively.

Figure 3.13 shows the histograms for the start and end time classifications for each algorithm, and Table 3.3 lists the summary statistics for both algorithms.

Algorithm 1 vs the Calendar

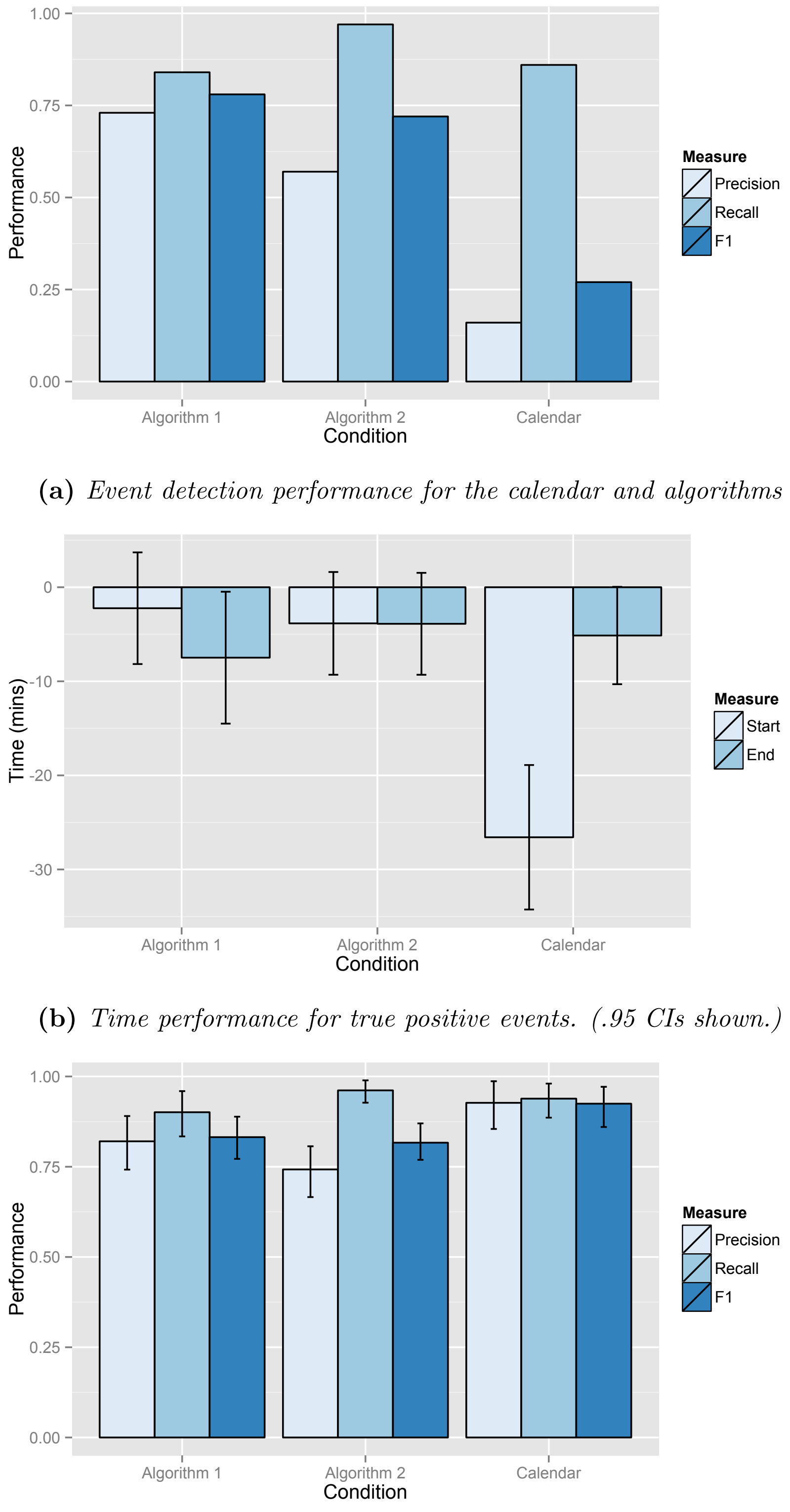
Comparing Algorithm 1 against the standalone calendar, there is a significant improvement in event classification precision ($p < 0.01$; exact Binomial test; 38 successes, 52 trials, hypothesised probability = 0.16). Comparing the statistics for the calendar events in G against the events in T, there is a large improvement in location F1 score, and there is a significant difference in start time estimation ($t_{70.085} = 5.0806; p < 0.01$;



(a) *Start time differences (mins)*

(b) *End time differences (mins)*

Figure 3.13: *Distributions of start and end time differences. Row 1 shows the standalone calendar events G relative to actual events; row 2 shows Algorithm 1 events relative to actual events; and row 3 shows Algorithm 2 events relative to actual events.*



(a) *Event detection performance for the calendar and algorithms*

(b) *Time performance for true positive events. (.95 CIs shown.)*

(c) *Attender detection performance for true positive events. (.95 CIs shown.)*

Figure 3.14: *Comparing performance between the standalone calendar and the data fusion algorithms.*

Statistic	**Algorithm 1**	**Algorithm 2**
Event precision	0.73	0.57
Event recall	0.84	0.97
Event F1	0.78	0.72
Start time $(\bar{x}_s, s_s)$	$(-2.23, 17.82)$	$(-3.84, 17.96)$
End time $(\bar{x}_e, s_e)$	$(-7.49, 21.62)$	$(-3.89, 18.31)$
Location precision	0.84	0.97
Location recall	1	1
Location F1	0.91	0.98
Mean attender precision	0.82	0.74
Mean attender recall	0.90	0.96
Mean attender F1	0.83	0.82

Table 3.3: *Summary statistics comparing the data fusion algorithm outputs with actual events.*

Welch two-sample t-test), with no significant difference in start time deviation from 0 ($t_{37} = -0.7609; p = 0.45$; Student's t-test). There is no significant difference in end time estimation ($t_{69.474}; p = 0.59$; Welch two-sample t-test) but there is a significant difference in end time deviation from 0 ($t_{37} = -2.1619; p < 0.05$; Student's t-test). There is no significant difference in attender F1 score ($p > 0.05$, non-parametric bootstrap 1000 replicates).

Algorithm 2 vs the Calendar

Comparing Algorithm 2 against the standalone calendar, there is a significant improvement in event precision ($p < 0.01$; exact Binomial test; 43 successes, 75 trials, hypothesised probability $= 0.16$) which, when coupled with high recall, results in an improved F1 score over the calendar. Comparing the statistics for the calendar events in G against the events in T, there is a large improvement in location F1 score over the calendar. There is a significant improvement in start time estimation ($t_{68.947} = 4.8807; p < 0.01$; Welch two-sample t-test), with no significant difference in start time deviation from 0 ($t_{42} = -1.4184; p = 0.16$; Student's t-test). There is no significant difference in end time estimation ($t_{79.951}; p = 0.74$; Welch two-sample t-test), with no significant difference in end time deviation from 0 ($t_{42} = -1.4467; p = 0.16$; Student's t-test). There is no significant difference in attender F1 score ($p > 0.05$, non-parametric bootstrap; 1000 replicates).

Algorithm 1 vs Algorithm 2

Comparing Algorithm 1 against Algorithm 2, Algorithm 1 appears to outperform Algorithm 2 for event F1 score. Comparing statistics between events in T for both algorithms, Algorithm 2 has better location awareness, though there is no significant difference in start time estimation ($t_{79.415}; p = 0.69$; Welch two-sample t-test) or end time estimation ($t_{73.856} = 0.8215; p = 0.414$; Welch two-sample t-test). There is no significant difference in attender classification F1 score ($p > 0.05$, non-parametric bootstrap; 1000 replicates).

3.5.9 Discussion

In this section, we discuss our findings and how they address RQ 2.1. We then analyse possible causal factors that affect performance.

RQ 2.1: To what extent does combining the calendar with other sources of context data affect overall sensing performance?

From the results, we see that Algorithm 2 outputs a greater number of true positives and fewer false negatives than Algorithm 1 but with a larger number of false positives. This results in a lower F1 score for Algorithm 2. Comparing these results to those of the standalone calendar, we see that data fusion reduces the number of false positives (from 204 to 32 for Algorithm 1 and 14 for Algorithm 2) and false negatives (6 for Algorithm 1 and 1 for Algorithm 2) to improve the F1 score from the standalone calendar.

From the results in Figure 3.14, it would appear that data fusion significantly improves context sensing from the standalone calendar. Both algorithms increase the event F1 score from 0.27 to above 0.7. This is because the additional context provided by the mobile devices and email contacts eliminate many of the false positive events in P and S. Also, because the calendar location field is rarely used (recall score of 0.11), the introduction of the Bluetooth location sensing significantly increased the location F1 score from 0.2 to above 0.9 for both algorithms.

We also see a significant improvement in start time awareness for both algorithms when compared with the standalone calendar. However, end time awareness for both methods do not significantly improve upon the calendar. Possible reasons for this are discussed shortly. Both algorithms' mean start time classifications contained the actual events' time within their 95% confidence intervals, and Algorithm 2's end time contained the actual events' end time within its 95% confidence interval. Algorithm 1, however, had a slightly poorer estimate of event end time than both Algorithm 2 and the calendar.

Interestingly, both Algorithms do not differ significantly from the calendar in attender classification performance. In fact, the results show that the calendar is actually a marginally better attender classifier than the algorithms, due in part to spurious attender classifications from participant 'walk bys' (discussed in detail shortly) – affecting precision – and Bluetooth failure – affecting recall. This would suggest that, once events are classified, using the calendar as the attender classifier may give improved performance.

Overall, it would appear that data fusion improves context sensing performance, and is therefore a better virtual context sensor than the calendar alone.

Which Algorithm is Best?

The answer to this question depends on the intended application. If event and attender classification performance were a priority, then Algorithm 1 would appear to be a better choice. However, if timing and location awareness were higher priorities, Algorithm 2 would be a better choice.

What Affected Performance?

Here the implications of both event and attender false positives and false negatives are discussed, followed by an review of their root causes.

- **False positives**: Depending on the type of application that uses calendar-based context sensing, false positives will vary in their significance. If privacy were a critical factor, then they would be very significant: we would not want users added to events that allowed them access to sensitive content intended only for participants in the event. In this case minimising false attender positives is imperative. Moreover, false event positives can be seen as a form of spam. Imagine a scenario where two users are walking past each other with a calendared placeholder. A false event may be created around this placeholder since the users are co-present, in each other's contact network and sharing a calendar 'event'. To the users, who in reality are not part of any such event, this could be irritating if, for example, the system attempted to remind them of the event or share media from the event with them.
- **False negatives**: Failure to identify attenders or entire events results in additional burden to users of such a system. If a user were not added to an event they were really part of, then they would have to be manually added. This could become tedious if failures are common. Failure to identify events can lead to further burden: users would have to create the event manually.

Here we present a cause and effect review of the false and failed identifications in our study. Table 3.4 lists the effects along with their likely causes.

Effect	Causes
False positive events	False sensor readings; participant mobility
False positive attenders	False sensor readings; participant mobility
False negative events	Sensor failure; false sensor readings
False negative attenders	Sensor failure
Event time deviation	Participant mobility; false sensor readings; sensor failure

Table 3.4: *Observed effects and their likely causes*

False sensor readings are sensors not reporting the true state of the world. Examples from our study include: (i) Bluetooth radio reflections causing devices to see each other when outside normal ranges of coverage, e.g. participants at their desks reported as 'in a meeting area'; (ii) Calendar placeholder events and shared reminders. We use the original calendar as a virtual sensor in the data fusion algorithms, and the large number of false events we saw in our analysis manifested as false sensor readings. Thus we saw examples where users were sighted as co-present at the same time as placeholders or shared reminders in their calendar. This greatly increased the chance of a false event or attender classification.

Sensor failure occurred when the sensors did not report data to the system when they should have. We observed the occasional Bluetooth sighting failure, i.e. participants not being sighted when in the ranges of coverage depicted in Figure 3.7. Occasional connectivity issues were observed when Bluetooth scan results were not reported in real time. Results that were not communicated were stored locally on the participant devices until a connection was re-established. However, in some cases, the results were reported after the event had occurred. It is possible to use this data to create the event *post hoc*, but real-time functionality is damaged.

In both data fusion algorithms, we requested calendar entries at one particular time (the time of execution), so entries listed near that time were not considered as possible candidates. We saw how variable the calendar time differences were, therefore it could be argued that introducing fuzzy time and requesting entries in a (perhaps weighted) time window could capture the calendar entries associated with such events, and help reduce the number of false negative events.

Participant mobility concerns the movement of participants around the study space. Even though we carefully chose the location of the static devices, we observed cases of participants moving into these areas when not involved in events there. An example of this was a participant who would frequently stand in a meeting area making calls on his mobile phone, which was being identified by the Bluetooth scans. Sometimes a relevant event was occurring in the meeting area, an attender of which had a social tie to this participant. The system therefore identified the participant as attending the event, resulting in a false attender positive. This problem also occurred when participants walked by meeting areas with ongoing events; the system would add them to the events if they had social ties to participant attenders.

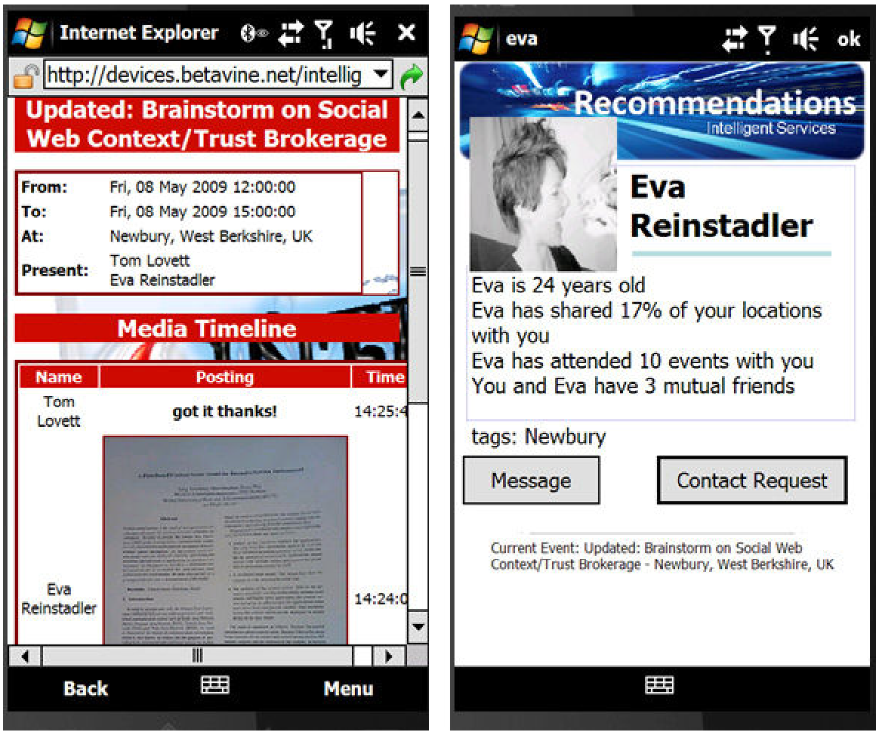
3.5.10 High-level Implications and Limitations

RQ 2.1 asks about the extent to which fusing calendar data with other context sensor data affects sensing performance, and RQ 2 – the high-level research question – asks about the extent to which data fusion affects context sensing performance in general. Our findings from this study have shown that, in this case, data fusion significantly improves overall performance, and that it allows useful data to be extracted from an otherwise noisy context sensor (the calendar). These findings have promising implications for the higher level research question; they show that fusing physical (Bluetooth location) and virtual (calendar and email contacts) sensors appears to be a worthwhile endeavour, and raises the question of which sensors might perform well together.

The key limitations of studying these particular sensors in this manner include: breadth of impact in relation to RQ 2 and a narrow scope for data fusion in general. For the former, we still need to address other combinations of context sensors and, for further work in order to strengthen the case for answering RQ 2, we could append useful *combinations* of sensors (and data fusion methods) to the context sensor catalogue proposed in Section 3.5.6. Moreover, our study could be easily repeated for various combinations, but this could be a laborious task given the number of possible sensor combinations ($\sum_{i=1}^{N} \frac{N!}{i!(N-i)!}$, where N is the number of sensors to test). For the latter limitation – data fusion scope – the immediate next step to address it would be to design and compare alternative data fusion approaches for the chosen sensor sets. This might include probabilistic approaches, e.g. Dempster-Shafer theory, or other logical ones such as ours. In addition to studying different sensor combinations, this would significantly strengthen the work for RQ 2.

3.5.11 Prototype Applications

During the course of this work, we developed a small set of prototype mobile applications that utilised knowledge of classified events to provide various simple services to a mobile user; particularly office workers. Implemented on Windows Mobile 6, these services are: a mobile photo sharing service, a people recommendation service and a mobile Twitter service.



Updated: Brainstorm on Social Web Context/Trust Brokerage

From:	Fri, 08 May 2009 12:00:00
To:	Fri, 08 May 2009 15:00:00
At:	Newbury, West Berkshire, UK
Present:	Tom Lovett Eva Reinstadler

Media Timeline

Name	Posting	Time
Tom Lovett	got it thanks!	14:25:4
Eva Reinstadler		14:24:0

Eva Reinstadler

Eva is 24 years old
Eva has shared 17% of your locations with you
Eva has attended 10 events with you
You and Eva have 3 mutual friends

tags: Newbury

Current Event: Updated: Brainstorm on Social Web Context/Trust Brokerage - Newbury, West Berkshire, UK

Figure 3.15: *Device screen captures showing the output of the event photo and Twitter sharing services (L) and an example recommendation from the event-based people recommendation service (R)*

Event-Based Mobile Photo Sharing Service

This service allows the user to take and upload photographs from her mobile device to her online Flickr and Facebook accounts. The photographs are time stamped and can be uploaded during or after an automatically classified event. The remote photo destinations (e.g. Flickr URLs) are stored and a web timeline is automatically created for the event with the event metadata attached; see Figure 3.15. The photos are displayed in the timeline and are viewable by users who are identified as part of the event. Thus, through this application, various media content associated with events can be captured and catalogued according to the event.

Event-Based Mobile Twitter Service

This service is modelled on the photo sharing service. The user can post status reports to her online Twitter account from her device. The remote destination of the status is stored, allowing the status to be retrieved from Twitter for display to the event attenders. As with the photo sharing service, the status reports can be added to the event timeline and displayed chronologically along with the photographs. Figure 3.15 shows an example timeline consisting of a photograph and Twitter status, with the event metadata displayed above.

Mobile People Recommendation Service

Designed for use both during an event and outside specific events, this service analyses various metrics output from the event history: co-location ratio (a measure of how often users are identified as being together), mutual contacts, personal data, shared events and auto-tags. Personal data includes information such as date of birth, taste in films, music etc. Auto-tagging creates common tags based upon event metadata and data from event attenders such as personal interests.

A list is produced of recommended people who are not currently contacts but frequently share user context i.e. 'familiar strangers' [177]. The user is able to view the matches and the reasons for recommendation on the mobile device; see Figure 3.15. She can make a request to the recommended contact and, if the contact accepts, a social tie is made between the user and contact. The details of the user and the contact are then added to each other's devices.

3.6 General Discussion

In this section, we discuss how the work in this chapter addresses our higher level research questions surrounding context sensing in mobile context aware systems, as well as the implications and limitations of the work.

3.6.1 RQ 1: What entities might we consider as virtual context sensors?

As we saw in Sections 3.2 and 3.3, almost anything that can provide data on context facets from people, devices and the environment can be considered as a context sensor. There are two broad sensor categories in context aware systems [15, 103]: *physical sensors*, which supply data about context sources to the system through physical interfaces, e.g. through an electronic signal; and *virtual sensors*, which supply data through virtual interfaces, e.g. software APIs. Virtual sensing is an interesting area for research due to the proliferation of web-based APIs and social media websites available to modern mobile devices.

We undertook a study that investigated the calendar's performance as a virtual context sensor, and from this we saw that – standalone – it is not particularly good at sensing context; mainly due to inherent noise, but due also to users not populating it with every significant event. The implications for this are twofold: (i) virtual sensors are only as good as the users who contribute the data to them, e.g. social media websites' sensing performance will likely vary according to user activity; and (ii) the entities that we are terming 'virtual sensors' are usually designed for other purposes other than sensing, e.g. the calendar for scheduling, and so we cannot expect to directly sense context using them. They are promising as practical context sensors however, as it costs little in terms of device resources to query them.

The key limitations of our work lie in generality. Firstly, our study only considered the office calendar as a context sensor. Users are perhaps more likely to use calendars in an office than in their personal lives due to the large amount of collaboration in the workplace, so the personal calendar may be an even worse context sensor than the office calendar. Secondly, we only analysed a single virtual sensor – the calendar. An interesting avenue for future work in this area would be to study other virtual sensors such as social media websites, and to analyse their context sensing performance. As we discussed in Section 3.5.6, further work to address these limitations might include a catalogue of virtual context sensors and their measures of performance as context sensors.

3.6.2 RQ 2: To what extent does combining multiple context sensors affect sensing performance?

From our study, we saw that data fusion generally improved context sensing performance when compared to single sensors. This has interesting implications for context sensing in general: first, with the abundance of physical and virtual sensors available to a mobile device, we could combine many of them through data fusion and analyse their overall sensing performance; and secondly, formal measures of context quality and reliability could be developed that could be used to standardise context sensors – thus, certain data combinations could be considered as sensors in their own right.

Of course, we only developed and analysed two data fusion algorithms and applied them to a single field study, but the results showed significant improvement. Perhaps a general approach to context data fusion could be developed and applied to further use cases in mobile environments. As we discussed in Section 3.5.10, further work to improve the generalisation of our work and better answer RQ 2 might be to catalogue useful combinations of fused context sensors and alternative methods of data fusion for each.

3.6.3 Implications and Limitations for Context

One question surrounding the work in this chapter is that of the theoretical understanding of context. During the calendar study, we focused on the full set of context facets, namely: Who (attenders), What (event type), Where (location), When (times) and Why (event names). However, although we covered a broad set of facets, we only considered a smaller set of instantiations of these facets. What are the implications for the theoretical understanding of context? The work shows that the Five Ws can be usefully applied to a real-world use case of context sensing, and that they capture a broad set of context data. Of course there are limitations for theoretical understanding: by limiting our instantiations of context to those provided by the calendar, we did not consider examples such as human emotion, intent or richer activity recognition. In doing this, there is still work to be done on advancing the understanding of context. Perhaps repeating the study with more sensors to capture facet instances such as fine-grained activity and integrating work in emotion sensing and intent inference, e.g. [186], could better improve the richness of context understanding.

3.7 Conclusion and Chapter Summary

In this chapter, we studied context sensing in mobile context aware systems. We began by specifying the requirement for context sensing, which led to a set of design questions: (i) what facets of context are we trying to sense?; (ii) from where can we usefully sense context?; (iii) what sensors are available to us?; and (iv) are there any constraints?. We then reviewed a set of potential context sources and sensors, before outlining the difference between physical (hardware) and virtual (software) context sensors. Key sources of context data are people, devices and the environment in which the people and devices operate [57].

We then explored how data fusion – in which multiple data sources could be combined in order to improve the accuracy or fidelity of the data beyond each individual source alone – could be applied to context sensing. This led us to consider the case of the office calendar as a virtual sensor of context, a case for which we chose to undertake a field study. The study resulted in two key concrete findings:

- The standalone office calendar is not a good context sensor, due mainly to low precision caused by a large number of events that do not actually occur in reality. Moreover, for the events that do occur, estimations of time are poor, though attender lists do appear to be accurate.
- By using data fusion to combine the calendar with other context sensors – namely

> social network and Bluetooth location data – we can improve context sensing significantly beyond the calendar alone; particularly for event classification performance and real time awareness.

These findings have interesting implications for presence and availability systems, particularly in a workplace environment. By employing our data fusion algorithms, workers can increase their belief in their colleague's purported presence and availability. Moreover, the underlying data fusion algorithms can enable unique applications that rely on the classification of events to provide a service to the end user, e.g. a meeting media and documentation capture tool. Furthermore, our approaches could assist in the purging of 'noise' from the office calendar, allowing companies and workers to automate clean-ups of their calendar databases.

Finally, the findings in this chapter have contributed to RQ 1 and RQ 2 by showing that we can consider multiple entities as physical and virtual sensors of context, and that combining multiple context sensors together is likely to improve context sensing performance.

This chapter concentrated on sourcing and sensing context in the sensing layer of our layer model, and the next chapter moves up to the intelligence layer; in which sensed context data can be processed for context inference and machine learning. The implementation of context inference and learning in a mobile environment comes with its own set of problems and challenges beyond those found in a more traditional desktop environment. The next chapter will address some of these problems and contribute new approaches to inference and learning in mobile context aware systems.

Chapter 4

Interactive Intelligence: The Intelligence

In the previous chapter, we studied context sensing in mobile context aware systems, which included a review of context data sources, common context sensors and the context facets to which they relate. We also showed how we might apply data fusion to context sensing, before reporting on a case study of the everyday calendar as a virtual context sensor.

In this chapter, we move up our layer model to the intelligence layer (see Figure 4.1), in which context data from the sensing layer and interaction data from the interaction layer are intelligently processed. Intelligence is the first component of *interactive intelligence* in mobile context aware systems. Here we integrate user feedback supplied from the interaction layer into the inference and learning processes of the intelligence layer. We address two areas of intelligence in this chapter:

- **Context inference**: inferring what state of context may have generated a set of observed – or sensed – context data.
- **Context learning**: learning from observed context data and supplied context knowledge in order to improve future inference performance.

As we saw in Chapter 2, context inference and learning – particularly in a mobile environment – are popular research topics in the UbiComp field. We aim to address the following two research questions in this chapter:

- **RQ 3**: To what extent can we infer significant changes in context using mobile devices?

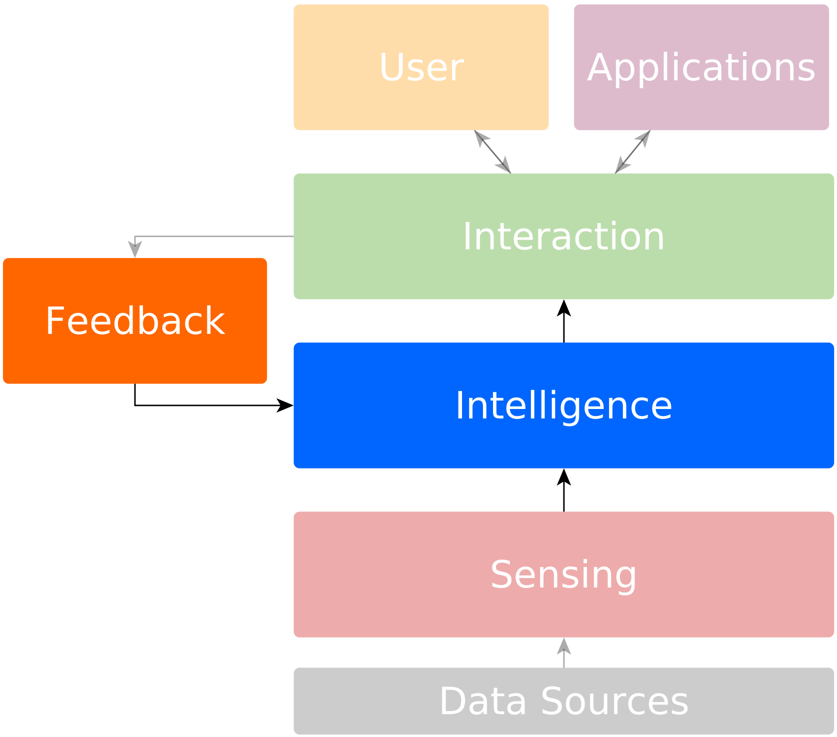


Figure 4.1: *This chapter will focus on the intelligence layer and introduce the idea of user feedback in mobile context aware systems.*

- **RQ 4**: To what extent can we infer and actively learn about context using mobile devices?

RQ 3 is concerned with the problem of knowing *when* to trigger sensing, inference and learning processes in a mobile environment. By considering context as discrete states in a finite state machine (FSM), can we identify the moment of state transition using mobile devices, and can we use these transitions as triggers for sensing, inference and learning processes? How well can context states be modelled by a FSM? For example, should travelling between places be considered a state or a transition? RQ 4 is concerned with real time context inference and learning with user feedback. How can we design context inference and learning algorithms that integrate user feedback in real time? Here we use active machine learning [210], a form of machine learning in which algorithms can query 'oracles' – users in our case – for feedback on their outputs.

To address these questions, we will focus on a relevant use case: *place awareness*, which is the awareness of *places* that are personally meaningful to people, e.g. "home" or "desk at work", rather than locations, e.g. an address or latitude-longitude coordinate. We will show how mobile device motion can be used to infer significant transitions between places (and how different parameters of our approach affect performance), and how we can incorporate user feedback into the place inference and learning process. Using active learning, we can achieve good place inference performance in real time with relatively little user feedback. This raises interesting questions about the amount of feedback we can realistically expect from users, and how we might attempt to elicit this feedback in a mobile environment.

We begin the chapter by providing an overview of intelligence in mobile context aware systems, including the requirement for the intelligence layer of our model. We also distinguish between context inference and learning, and discuss how we might design for interactive intelligence; which incorporates user feedback into the inference and learning process. By considering context as a series of discrete states in a FSM, we design a high level context inference and active learning algorithm that is controlled by the transitions between the FSM states. Following this, we introduce the place awareness problem in detail, and outline our rationale for choosing it with regard to RQ 3 and RQ 4. We narrow down these research questions to the scope of place awareness, and present the results from two hybrid laboratory/field studies that were designed to address the questions. Finally, we reflect on the implications and limitations of the work with regard to the research questions, and link it to the work in the following chapter about the interaction side of interactive intelligence.

4.1 Intelligence in Mobile Context Aware Systems

In this section, we provide an overview of intelligence in mobile context aware systems by defining what we mean by intelligence, and outlining the requirement for intelligence. Intelligence, in the philosophical sense, is the ability to acquire and apply knowledge and skills[1]. Within AI, Russell and Norvig define an intelligent agent to be "*anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.*" [199].

Combining these definitions for context aware systems, intelligence is the ability to acquire and apply knowledge of context given data obtained through context sensors. The key element is context knowledge *acquisition* – how can our system *know* or reason about context? This can be broken down into two problems: (i) context inference: given a set of observed – or sensed – data, can our system reason about what context may have generated the data? (ii) context learning: can our system *learn* from experience in making inferences?

Context inference draws conclusions about context based on a combination of observed

[1] "intelligence". Oxford Dictionaries. April 2010. Oxford University Press. `http://oxforddictionaries.com/definition/english/intelligence` (Accessed 2012-11-05)

data and prior knowledge, and context learning is concerned with obtaining the prior knowledge. Learning is about adaptability and improving future inference performance given observations over time [199] and context, by its nature, is certainly dynamic [80].

4.1.1 Context Inference in a Mobile Environment

In statistical nomenclature, context inference can viewed as a classification problem, i.e. how can we assign a meaningful label or category to a context data observation? Context inference is a large sub-field within context awareness research and – as we saw in Chapter 2 – a large amount of work has been conducted on inference alone in recent years. Within the field of mobile context awareness, context inference using mobile devices poses a variety of interesting problems for researchers. For example, can we design a context inference process to be as accurate, responsive and resource-efficient as possible given the various constraints posed by mobile devices, e.g. CPU speed and battery power? When designing for good inference in a mobile context aware system, we may often have to deal with requirements that compete with each other, e.g. context inference accuracy *vs* resource efficiency [186].

As we saw in the previous chapter, context data can be obtained from a variety of physical and virtual sensors. In a mobile environment we have the added property of mobility, where users interact with their mobile devices in variable, dynamic and often unpredictable situations.

Inferring the context of mobile device users is valuable but typically non-trivial in practice. If we are to make useful and relevant inferences about context, we should not only acquire rich data from available context sensors, but we should also take advantage of the mobility aspect and utilise input from the users themselves.

4.1.2 Context Learning in a Mobile Environment

There are two important types of machine learning [27]: *supervised learning* – where the learning process is shown some training data and attempts to learn a function that predicts outputs based on unseen test data – and *unsupervised learning* – where the learning process attempts to recognise certain characteristics about the input data without assistance.

In a mobile environment – unlike many desktop environments – we rarely have the luxury of large training data sets with which to train supervised learning algorithms. Researchers have side-stepped this issue by training models offline, i.e. by gathering data from the device and sending it to a remote desktop machine to train, before sending the trained model back to the device for context inference [98, 154]. This does not completely solve the problem however, as training data need to be labelled, communicated remotely, processed and sent back to the device, which could introduce latency and financial cost into the inference and learning process.

Although data is becoming cheap and connectivity is improving, there is potentially little latency and financial cost to executing context learning processes – particularly unsupervised ones – on the device itself. Indeed, with increased smartphone processing capabilities, this seems very appealing, but it introduces another cost: energy. Turning on sensors and running sophisticated context learning algorithms is likely to have a severe impact on mobile device battery power [118]. This could further impact on how well context learning algorithms and, consequently, inference algorithms perform [186].

This is the crux of the context learning problem: how can we effectively learn about users' context over time whilst preserving mobile device usability? Given a potential lack of training data, could we encourage the user to train their device about their context with minimal burden?

4.1.3 Eliciting and Incorporating User Feedback

Eliciting input data from users in mobile environments is a well-studied problem in HCI. For example, the Experience Sampling Method (ESM), e.g. [51] – where users are encouraged to input feedback data about their experiences into a device – is a popular approach to eliciting user data in the field. Various methods have been presented in recent years to balance requests for user input against automated inference processes. Rosenthal, Dey and Veloso [196] use a decision-theoretic learning approach to experience sampling for triggering mobile interruptions, where context is initially sampled periodically to build a learned model of user interruptibility. Kapoor and Horvitz [112] compare and contrast a range of interruptibility sampling methods, including a sophisticated decision-theoretic approach that builds a predictive sampling model of the user. These are good examples of using both machine intelligence and occasional user input to improve the inference performance of context aware systems in the field.

In the mobile environment, the MyExperience platform [73] and its applications, e.g. [74], sample both subjective context, i.e., user experience, and objective context, i.e., sensor data. The sensor data is logged periodically and used as an event-based trigger for experience sampling. Mobile event-based sensor sampling methods are used in both the AndWellNess [88] and EnTracked [119] systems for fitness based experience sampling and position tracking respectively. Furthermore, event-based sampling using mobile device accelerometers and cellular events is used to good effect for determining transportation modes by Reddy *et al.*[191], and work by Ho and Intille [92], use changes in motion using body-worn accelerometers to infer transitions between users' different activities.

Incorporating user feedback into the automated inference process in real time can be seen as a form of active machine learning [210]. Active learning involves the user directly in the learning process [3] with the intention of improving future inference performance by telling the system whether it is right or wrong. For context learning, active learning techniques have been used in a desktop environment, e.g. [112, 196] and, to some extent, in a mobile environment using wearable sensors, e.g. [125, 217].

4.1.4 The Intelligence Requirement

Following the Five Ws model of context facets, we can outline the key questions and operational requirement for intelligence in a mobile context aware system:

1. **Who**: how might we infer and learn about *identity* from sensed context data? This could be the identity of a device user, or the users' friends.
2. **What**: how can we infer and learn about what the user or device is doing, i.e. the *activity*, from sensed context data? This is what activity recognition is primarily concerned with.
3. **Where**: can we infer and learn *spatial* information from the sensed context data? Due to its increasing popularity in the mobile environment, location-awareness has become reasonably trivial to implement within services and applications. There are limitations to location-awareness, however, particularly indoors where GPS is unlikely to work well. Our goal here may depend on the application requirements – do we need to know location to a fine degree of accuracy? Are we inferring a location, address or perhaps the meaning of a location?
4. **When**: how can we capture *temporal* patterns from sensed context data? More importantly, can we tell if the data we are observing are *relevant*? For some applications, data that are hours or days old may be deemed relevant; but for others, data may need to be supplied as close to real time as possible.
5. **Why**: can we infer and learn about context *meaning*, e.g. the name of a location, user emotion or the relationship between the user and another person? This is perhaps the most non-trivial context facet to infer and learn about [187].

The intelligence requirement for a mobile context aware system – following Dey's definition [57] – is to *infer and learn about the information that characterises the situation of the relevant entity.* Though simple to state, satisfying this requirement in practice is non-trivial, especially in a mobile environment.

4.2 Context Inference and Active Learning

In this section, we outline an abstract approach to context inference and learning in mobile context aware systems which incorporates user feedback. By first considering context as a FSM, we develop a high level algorithm that performs real time context inference and incorporates user feedback into the context learning process. This algorithm will be used in our case study of place awareness in the next section.

4.2.1 Modelling Context for Inference and Learning

As we saw in Chapter 2, context has been classified and modelled using a variety of means. Unfortunately, there is no *de facto* standard model of context, as most models are manifestations of a design process and are therefore specific to the application they were designed for. There are notable approaches to general modelling, e.g. Dey and Abowd's set of criteria [59], and models range from the abstract [109] to the formal [25], but model choice is still guided by application in much of the literature.

In this chapter, we are interested in modelling context specifically for inference and learning. In order to do this, we must identify what elements of the inference and learning problem can be best represented by which model, and what the implications of our choice might be for other elements of the problem.

In the next section, we outline a rationale for using a finite state machine model of context for inference and learning. We also outline its advantages and disadvantages when used to model context in this manner, and reflect on its comparisons with other alternative approaches.

4.2.2 Context as a Finite State Machine (FSM)

A FSM is an abstract model of a system, in which the system is in a particular state – one of a finite set – at a given point in a process (or time). FSMs have been used to model context in previous work, e.g. [231], as they effectively 'discretise' the otherwise abstract and dynamic process of context. Although abstract, FSMs do allow us to capture some important and useful properties of context in a systematic way.

An example FSM is shown in Figure 4.2. Here, each of the discrete context states are reachable from a certain subset of the others through state *transitions*. We are implicitly implying that state transitions are a temporal process and, as such, there could be context states that our system has not yet observed or our user has not yet experienced or defined. This is not easy; as Greenberg notes [80]: "*Determining an*

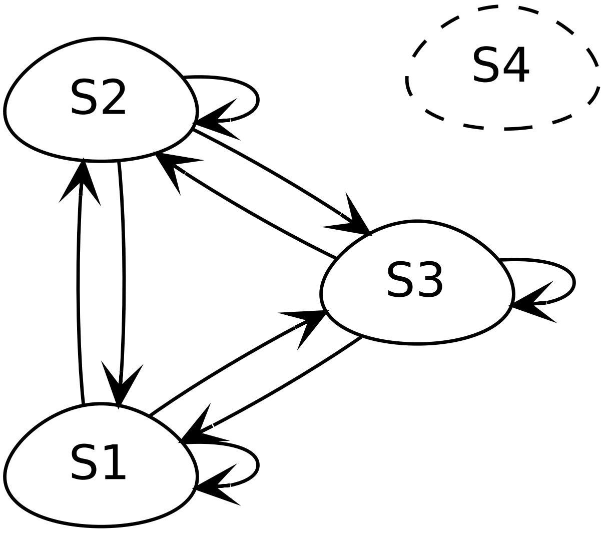


Figure 4.2: *Context as a FSM, showing transitions between states and an as yet unobserved state (S4).*

appropriate set of canonical contextual states may be difficult or impossible. Figure 4.2 illustrates the possibility of a self-transition, an example of which might be a person walking away from their desk and returning to it having forgotten something.

Why then, is an FSM a suitable model for context inference and learning? Here we list its advantages and disadvantages – comparing them with alternative models where appropriate. Some of the key advantages of using an FSM in this manner include:

- An FSM can **discretise** and simplify what is a complex process of context dynamics over time [80]. Much like quantisation in digital signal processing, a 'continuous' signal can be mapped to a 'discrete' space for further processing at the cost of minor information loss. By considering context as a state-by-state process, we can use its discrete nature for applications such as mobile device ringtone profiles (e.g., putting the phone into silent mode for the duration of the state), notification delivery or application starting/stopping.
- An FSM can model **temporal flow** very well. Whereas alternative models capture static abstractions well, e.g. [25, 59], they do not model temporal processes adequately. As such, the 'When' facet of context is often neglected in favour of the spatial 'Where' or identifier 'Who'. As we are interested in *relevant* and *reactive* context awareness on mobile devices, time becomes very important – particularly the operation of sensing, inference and reaction in real time. By using an FSM, we can model context states over time; allowing us to monitor transitions between states as changes in context over time.

- An FSM is **event-driven**, i.e. state transitions are triggered by some pre-defined criteria. This is particularly attractive for mobile context aware systems as expensive sensing and inference processes can be event-triggered rather than polled, thereby saving energy and processing resources on the device. Of course we must carefully choose the criteria for transition as certain events may be missed [186], but once done we can effectively 'assume' that the current context state is constant until the next transition. Thus we could, for example, execute all our sensing and inference routines at the moment of transition, and turn them off until the next transition.

- An FSM is **abstract**. We define the states and the transition criteria, which allows for a large and diverse range of FSMs that can be tailored towards certain applications without being defined by them as other context models are, e.g. [73, 154]. There are disadvantages to this informality, as we discuss below, but the chief advantage is its adaptability.

We should note, however, that using an FSM does have potential disadvantages, including:

- An FSM does not model **concurrency** very well. Assuming a user is in one particular state may simplify the process of context over time, but it does so at the cost of losing subtler elements of context such as the notion that users can be in different context states simultaneously. This is arguably possible, e.g. watching TV whilst typing a document on a laptop, and not well modelled by the single-state over time approach of an FSM. This could be avoided by using multiple FSMs concurrently, or modelling a single state as a composition of more atomic ones, but this can become complicated, and the added complexity may outweigh the perceived advantages of using the FSM in the first place.

- Similarly, an FSM does not model **multi-faceted** context well. What defines a context transition? If we use the Five Ws, should a change in location (Where) be a transition, even though the activity (What) may not change, e.g. driving? What about more abstract context changes such as a change of emotion or intent (Why), even if activity or location remain constant? Unless modelling a single facet, an FSM may not be an ideal model of implicit context changes over time.

For the purposes of our work in this dissertation, the FSM's advantages outweigh its disadvantages. The event-driven discrete nature of the FSM fits well with most probabilistic inference and learning methods, e.g. probabilistic graphical models, due to the mapping of events in probability event space to states in the FSM. Moreover, the modelling of time is especially important for designing interactive intelligence –

where the implications of user interaction may be different depending on *when* the user interacts with the intelligence. Concurrency and multi-faceted context changes are perhaps a little beyond the scope of this work, though they are certainly grounds for future work given the research in this dissertation.

Given our choice of using an FSM to model context for inference and learning, we have three problems:

- How do we know *when* the context state transitions occur? What defines a transition?
- What characterises a context state? How do we recognise or infer it?
- How can we introduce new states as they are observed or defined?

The first two problems involve inference and learning, and link to RQ 3 and RQ 4. The third – which is linked to RQ 5 – is perhaps more complex. Without someone telling the system about new states or their characteristics, it may be extremely difficult to infer the identity or meaning of new states on first observation (though it may be somewhat easier to infer whether the state is simply new). This is Bellotti and Edwards' argument [20]: "*There are human aspects of context that cannot be sensed or even inferred*", and it is where the interaction component of interactive intelligence fits in.

With these problems in mind, we develop a general context inference and learning algorithm that incorporates user feedback into the inference process. The algorithm will then be applied to the case of place awareness in a series of user studies.

4.2.3 Context Inference and Active Learning Algorithm

The general idea behind active learning is to allow the learning process to choose the data from which it learns [210], with the intention of improving future inference performance and learning efficiency [149]. The key component is the query strategy used by the algorithm, i.e. what criteria are used to select the data for learning.

For our context inference and active learning algorithm, we use a measure of certainty (or uncertainty [134]) in the context inference. So, if our algorithm is uncertain about its inference, it will actively query the user for feedback on whether the inference is correct.

How do we reach this stage? First, we need an inference subroutine that can probabilistically infer the context state and can pass these inferences to the active learning component. In the general sense, this subroutine need only return a decision about

Algorithm 4 Context inference and active learning (executed upon transition into a new state)

```
**Input: set of previously observed context states $S$, their features $F$ (indexed by state) and a confidence threshold $t$.**
**Output: the set of states ranked by probability $\hat{S}$; the inferred state $s \in S$ and inference confidence $c \in [0, 1]$; an updated set of states $S$; and their features $F$**
$\bar{\mathbf{x}} \leftarrow$ ObserveContextData()                      ▷ Sensing.
**if** $\bar{\mathbf{x}}$ is empty **then**
    **return**
**end if**
$\hat{S} \leftarrow$ InferContext($\bar{\mathbf{x}}$, $S$, $F$)                ▷ Inference.
$s \leftarrow \hat{S}[1]$                                 ▷ The most probable state.
$c \leftarrow$ MeasureConfidence($\bar{\mathbf{x}}$, $S[s]$)     ▷ A confidence or certainty measure.
**if** $c < t$ **then**
    NotifyUser($\hat{S}$)                       ▷ Query the user if not confident.
**else**
    UpdateFeatures($\bar{\mathbf{x}}$, $F[s]$)                        ▷ Learning.
**end if**
**return**
```

what context may have generated a data observation. Of course, we need to observe data to do this, which means our sensors need to supply the data. Where do we start? How do we bootstrap this entire process?

This is where context transitions come in. We have seen that, by using an FSM to model context over time, context states are effectively static until the moment of transition into another state. Ideally, if we observe data at the moment of transition, we shouldn't need to take another observation until transition into the next state. This is particularly appealing for mobile context aware systems, where turning on sensors impacts on battery resources.

We therefore have the key components for our algorithm:

- A bootstrap, or trigger: the moment of transition into a new context state.
- Context inference: a probabilistic measure over each previously observed state, along with a method for concluding the current state.
- Active learning: choosing which data to learn from, and when to query the user.

The algorithm is displayed formally in Algorithm 4.

There is a final component to the active learning process – what to do when the user *does* respond. At the abstract level, the user should only be able to do one of two

Algorithm 5 User feedback algorithm (executed upon user response)

```
 1: Input: the inferred state $s_b$ before user feedback, its confidence measure
    $c_b$, confidence threshold $t$, the state $s_a$ after user feedback, i.e. the ‘oracle’
    answer, $\bar{\mathbf{x}}$ the context data for the state, $S$ the set of existing states and
    their features $F$
 2: Output: the inferred state $s \in S$ with confidence $c$, an updated set of
    context states $S$ and their features $F$
 3: if $s_a \neq s_b$ and $c_b > t$ then          ▷ If confident and wrong, undo mistake.
 4:     ReverseFeatureUpdates($\bar{\mathbf{x}}$, $F[s_b]$)
 5: end if
 6: if $s_a \in S$ then                            ▷ Update knowledge of known state.
 7:     UpdateFeatures($\bar{\mathbf{x}}$, $F[s_a]$)                        ▷ Learn.
 8: else
 9:     $F_a \leftarrow$ GenerateFeatures($\bar{\mathbf{x}}$)       ▷ New state: characterise it.
10:     add $F_a$ to $F$
11:     add $s_a$ to $S$                           ▷ Add new state to the observed set.
12: end if
13: $s \leftarrow s_a$
14: $c \leftarrow 1$                                ▷ ‘Oracle’ confidence set to 1.
15: return
```

things: (i) tell the system it is correct; or (ii) tell the system it is incorrect and supply the correct answer. If the user tells the system that it is correct, it should use the data to continue its learning process. If incorrect, the system should attempt to undo any erroneous learning before relearning using the correct answer. This algorithm is shown formally in Algorithm 5; it is bootstrapped by the user response. It should be noted that the confidence threshold input to both algorithms must be carefully chosen, e.g. through cross-validation learning or using standard confidence intervals, in addition to the confidence measure, e.g. statistical confidence or information theoretic measures.

The ‘ReverseFeatureUpdates’ routine on line 4 of Algorithm 5 is designed to recover an erroneous feature update as a result of an incorrect inference. It works by performing the inverse operation of the ‘UpdateFeatures’ routine, therefore the feature updates should be mapped in order to perform this inverse operation. There are limitations to reversing the feature updates: it can only be applied to the most recent state update, and may not be viable if the corrected inference occured some time in the past. This is because the features may have been subjected to further updates since the erroneous update, and it may not be able to reliably recreate the effects of these intermediate updates.

In this section, we have developed an abstract algorithm for context inference and active learning. In the next section, we apply it to a concrete use case in order to demonstrate its functionality and analyse its performance in the real world.

4.3 Use Case: Place Awareness

In order to use a concrete basis for study, both this and the following chapter will focus on the use of mobile devices for place awareness. A 'place' is much more than a 'space'; as Harrison and Dourish propose [86]:

"*Physically, a place is a space which is invested with understandings of behavioural appropriateness, cultural expectations, and so forth. We are located in 'space', but we act in 'place'. Furthermore, 'places' are spaces that are valued. The distinction is rather like that between a 'house' and a 'home'; a house might keep out the wind and the rain, but a home is where we live.*"

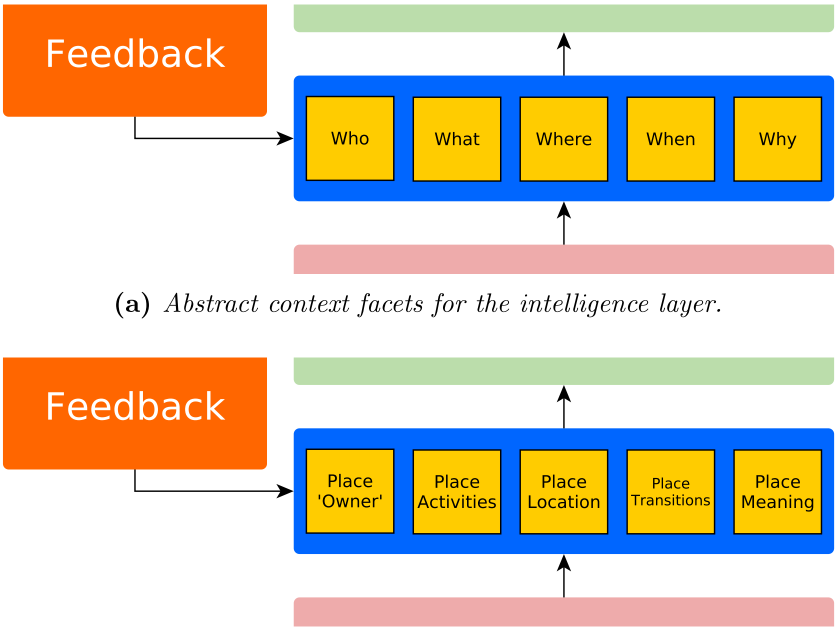
Following this definition, a place is a *physical area of meaning to a user*. Rather than a set of coordinates or an address, a place can be much more loosely defined with rich and often complex meaning. As such, the *subjective* nature of place awareness offers a richer and more personal user experience than the more *objective* idea of location-awareness. Indeed, as Barkhuus *et al.* show using a field study of their Connecto system [17], place descriptions can be broken down into four key categories of meaning:

- Geographic labels: similar to location or address.
- Place names: meaningful names, e.g. shop names or 'gym'.
- Activities: verb descriptors, e.g. shopping.
- Hybrids/expressions: sentences or combinations of the above categories.

Of these categories, the place name category was by far the most commonly used among the study participants, adding further empirical validation of Harrison and Dourish's original place *vs* space argument [86].

Why is place awareness a good case for studying interactive intelligence in mobile context aware systems? Places have clear personal meaning to people, but their loose, variable and often ambiguous descriptions present an inference and learning problem that goes beyond simple location awareness (the intelligence) [89]. Places have meaning, and it is difficult for intelligent systems to infer this meaning without users explicitly telling it to them (the interaction) [115]. Moreover, place awareness has practical applications for mobile service personalisation, e.g. predicting traffic between work and home; sharing personalised context data with friends and family; or delivering reminders when the user is at their desk or another relevant place. Place awareness is an active area of research in the UbiComp community [89, 90, 115, 116, 117, 133, 156].

In this chapter and the next, we engineer a novel mobile place awareness system that infers and learns about users' meaningful personal places in real time using their mobile



(a) *Abstract context facets for the intelligence layer.*

(b) *Concrete context facets for intelligence in place awareness systems.*

Figure 4.3: *Our layer model, showing the context facets for a place awareness system.*

devices and active learning. The system will attempt to capture place meaning by querying the user for feedback on its inferences, and it will attempt to learn about places over time using this approach.

In the next section, we outline the low level research questions for inference and active learning in mobile place awareness.

4.3.1 Related Work and Research Questions

Here we provide some background to place awareness with mobile devices, state our research questions and directly contrast our contributions for the remainder of this chapter against relevant work in the field. The problem of place awareness has received attention from researchers in recent years due to the ubiquity of mobile devices that can enable such awareness in people's everyday lives. There are two general approaches to place awareness: geometric-based, where spatial coordinates are used for clustering into places; and fingerprint-based, where signatures (typically RF signatures) in the environment are used to identify place 'zones'.

Notable geometric-based systems include: Askbrook and Starner's GPS-based work [11], which clusters GPS coordinates *post hoc* to learn users' significant locations; Kang *et al.* [110], who use a time-based approach to cluster GPS coordinates and extract places in an *ad hoc* manner; Nurmi and Koolwaaij [164], who use online GPS coordinate clustering; and Liao *et al.* [137], who use supervised learning to identify place and activity transition sequences *post hoc*. The main disadvantage of these systems, however, is their dependency on GPS, which means they do not work well for finer-grained indoor places.

Notable approaches that use fingerprinting include: Hightower *et al.* [90] and Kim *et al.*'s place awareness work using wireless RF fingerprinting [116], SensLoc [117] and Loci [115]. These later systems operate well indoors, and use *device motion* averaged over short time windows to trigger wireless sensing in a fairly simplistic manner, i.e. through the use of apparently arbitrary motion and window time parameters. However, no analyses of how well genuine place transitions can be extracted from these motion data are performed. Similarly, Chon, Tapilov and Cha [42] and Chon *et al.* [41] use time-averaged mobile device motion as indicators of place transitions but, again, arbitrary parameters are used and no analyses of how the choice of parameters affect transition inference performance are performed. This is also the case in other mobile context aware systems that use motion-triggering for sensor activation e.g. [30, 119].

Kim *et al.*'s study of user feedback for place capture [115] is also one of the first place awareness approaches to consider incorporating user feedback for improving place inference and learning. Although the authors simply capture *in situ* feedback and do not integrate it back into any learning process, they do note the potential value of such interactive intelligence for place awareness.

The challenges and benefits of implementing interactive intelligence (IIS) have recently been studied by Acid *el al.* [3] and Stumpf *et al.* [218, 219, 220], and applied in the field for experience sampling by Rosenthal *et al.* [196]. This work in particular shows that utilising user feedback is useful for improving future inference performance through machine learning, but the process of eliciting feedback is challenging, particularly in a mobile environment.

Given existing work, how might we contribute to RQ 3 and RQ 4 using place awareness as a use case?

RQ 3.1: To what extent can we infer significant transitions between users' meaningful places using mobile device motion?

RQ 3 asks how we might infer significant context changes with mobile devices. The temporal nature of the problem, i.e. identifying *when* in time context changes occur, relates to the "When" context facet (see Figure 4.3a). By applying our FSM model to place awareness, the context transitions take the form of place transitions. So, how do existing approaches to place awareness infer place transitions, and how might we do better? As we saw in the related work, the use of time-average device motion is a common method for doing this, but – as far as we are aware – no one has analysed how best to choose these parameters, nor how their choice might affect place transition inference performance[2]. Choosing motion and moving average parameters that are too low risks unnecessarily turning on sensors and executing inference and learning algorithms which may impact on device battery life but, if they are too high, we risk missing transitions.

Our research question can therefore be scoped for place awareness – to what extent can we infer significant transitions between places using device motion? In other words, how does the choice of parameters for the commonly used smoothed motion triggering approach affect transition inference performance? The user study that addresses this question is detailed in Section 4.4.

RQ 4.1: To what extent can we infer and actively learn about users' meaningful places using mobile devices?

RQ 4 asks about context inference and active learning. For place awareness, this mainly involves inferring and learning about the "Where" and "Why" context facets (see Figure 4.3a) from available sensor data. How do existing approaches do this, and how might we do better? As the related work shows, almost all place inference and learning approaches – both geometric and fingerprint – prioritise *automation*, i.e. they are concerned with automatic identification of the "Where" facet without consideration of the "Why". Thus, places are not labelled with anything beyond a unique ID unless users manually add labels themselves; recent attempts at prompting for labels are limited to daily surveys [115] where users are unlikely to remember places accurately *a posteriori.* As Barkhuus *et al.* showed, place labels are extremely important to users [17], and this lack of labelling in existing approaches potentially limits the usefulness of mobile place awareness.

Our research question can be scoped as follows: how might we infer users' meaningful places – including their meaning – and actively learn about them? The novelty will lie in active learning, i.e. incorporating user feedback into the place inference and learning process. The user study that addresses this question is detailed in Section 4.5.

[2] There is a brief analysis of how varying moving average time parameters affects place inference performance in [117], but not transition inference performance.

What About the “Who” and “What”?

The reader may have noticed that the “Who” and “What” context facets are not considered. Like other approaches to place awareness, we are initially concerned with a single user’s collection of meaningful places rather than any collective interpretation of a place. This may be a fruitful area for future research, but for parsimony in this work we limited our study to the places of individuals.

The “What” facet is concerned with activity recognition and, although we should be aware of user activity within places (particularly when related to transition inference through device motion), it is a popular area of research that is beyond the scope of this work.

4.4 Study: Inferring Place Transitions

In this section, we present a user study that addresses RQ 3.1 by analysing how mobile device motion from the accelerometer can be used to infer place transitions, and how the parameters of this approach affect inference performance. In this study, we focus solely on capturing the moments of place transition (rather than inferring the places themselves) using mobile device motion data from the accelerometer.

One of the most challenging aspects of context inference in mobile systems relates to time. Ideally, context inference should be *reactive*, i.e. with minimal inference latency, and context aware services and applications should operate in real time. Any noticeable delay between a user entering a context state and the system becoming aware of the state is likely to impact on user experience and system performance. However, energy and processing constraints – particularly in mobile devices – can limit the sophistication of the inference techniques used to enable real time context awareness.

In the case of place awareness, real time awareness is extremely desirable, as many services and applications such as notification or experience sampling tools can benefit from ‘event-based’ triggers such as a user entering or exiting each place. Furthermore, context transitions can act as triggers for resource-intensive sensors or user notifications [92, 214]

As we saw in the previous section, no one has systemically analysed how place transitions could be inferred from device motion. This study contributes a systematic analysis of a place transition inference system that uses mobile device motion sensed by the accelerometer. More specifically, we analyse two factors – moving average time windows and weighting methods – and show that they have significant effects on transition inference performance. We begin by describing a systematic approach to place transition inference with mobile device motion. We then outline the design of a hybrid laboratory/field study that captures users' natural transitions between places in addition to high-precision observations, before reporting the results of our analysis and discussing how they address RQ 3.1.

4.4.1 Approach

In this section we outline our approach to inferring transitions between users' meaningful places using mobile device motion. First we outline the specific problems involved in transition detection from mobile device motion, before using them to inform our system design. We then describe our hybrid laboratory/field study in which we collect the necessary data for *post hoc* analysis of how the factors in our design affect transition inference performance. We begin by summarising the key problems involved in inferring the moments of place transition from mobile device motion data:

1. **Sources of motion**: device motion may be a manifestation of noise or a less significant activity, e.g. the user idly playing with her device, rather than motion associated with movement between places. Conversely, device motion may not always reflect user motion, e.g. the user leaving her device on a desk.
2. **Subjectivity**: the intensity of motion that indicates a state of 'stationary' or 'moving' may vary between states and between users.
3. **History**: motion at a single point in time, or over a short period of time, may not provide enough information about whether the user is actually moving between places or not. Conversely, increasing the amount of historical data to be considered could affect inference performance and latency.

These problems provide a rationale for our design. The requirement for the transition inference system is – in real time – to broadcast a message when the user is transitioning into (entering) or out of (exiting) a place. Figure 4.4 shows the components involved in the system, and the following subsection describes its design.

System Design

The transition capture process is designed to operate on-device and uses two components commonly found in mobile motion detection systems: a logistic function and moving average time window.

- A **logistic function** addresses problem 2 – subjectivity. It is very unlikely for motion intensity to be consistent both within and between places, and motion

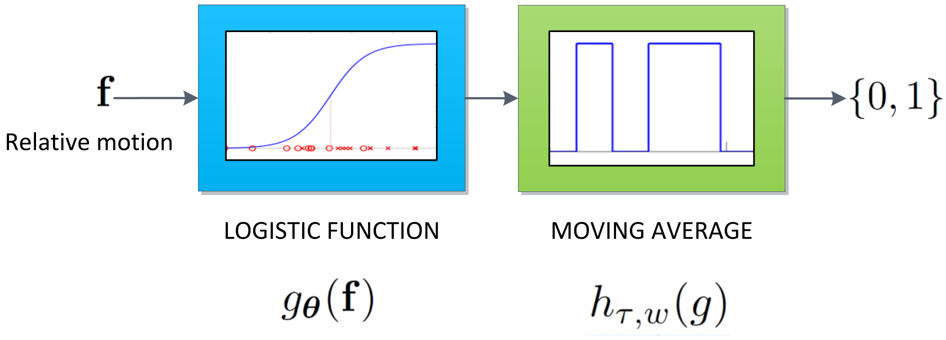


Figure 4.4: *The process design*

patterns will vary between users and the device's on-body location. The function outputs values in $[0,1]$ that represent the probability of the device undergoing significant motion (or the complement probability of insignificant motion) at timestep k, given the motion observed relative motion vector $\mathbf{f}$:

$$g_{\boldsymbol{\theta}}(\mathbf{f}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathrm{T}}\mathbf{z}}} \tag{4.1}$$

Where $\mathbf{f}$ is the relative motion vector at timestep k and $\mathbf{z}$ is a column vector in $\mathbb{R}^2$, defined:

$$\mathbf{z} = \begin{bmatrix} 1 \\ ||\mathbf{f}|| \end{bmatrix} \tag{4.2}$$

Here, $\boldsymbol{\theta}$ is a parameter vector in $\mathbb{R}^2$ that controls the function shape. We can learn these parameters for a specific user through regularised logistic regression on a sample of the user's motion data in various context states.

- **A moving average function** addresses problems 1 and 3. To minimise the effect of transient and unimportant motion, we can smooth the logistic function outputs over a fixed time window, τ, so that only sustained motion can trigger a transition. This uses historical data and, as such, requires a necessary lag to operate. In addition to varying τ, we can use weighting methods, w, for the historical data, which can vary the influence from more recently acquired data. The moving average outputs high if the weighted average over the logistic function outputs in τ is ≥ 0.5, and low otherwise. This threshold is chosen for the average of the logistic function's output over w, which in turn is learned using regularised logistic regression from a set of training data. Thus the threshold at 0.5 gives a decision boundary for average learned belief of motion-triggered transition.

Study Design

In order to evaluate system performance, we should design a study that will allow us to capture the data required for analysis in the most natural way possible. We have two goals: ecological validity, i.e. capturing data that is representative of the real world; and reliability, i.e. high-precision observations. A field study would satisfy the ecological validity goal, and a laboratory study would satisfy the reliability goal, but neither can easily satisfy both.

For place transition inference, we conducted an empirical study of mobile users in a hybrid laboratory/field study. Participants were asked to visit a set of meaningful personal places in a natural order, whilst being shadowed by a researcher recording what actually happened. We recruited 14 participants (11 male and 3 female; aged 20–38, mean age 27) from three different daily environments: an office (7 participants), a university (6 participants), and a town centre (1 participant). These environments were chosen to vary the movement patterns between participants and to lessen the effect of behavioural bias associated with all participants being located in the same environment. We recruited a mixture of male and female participants in order to record device motion that may vary between them, e.g. female participants carrying their device in their handbag rather than their pocket.

In a pre-study interview, we asked them to describe their typical day's activities chronologically through transitions between meaningful personal places within their environment. Each participant was told the difference between a place and a space using the Harrison and Dourish example of 'Home' [86] and a verbal explanation of example places given by Barkuus *et al.* in their Connecto system study [17]. Immediately following the interview, we asked them to choose a sequence of these places (and activities) that could be performed as a scripted tour. Each participant was equipped with an Android mobile device containing an accelerometer, from which the output was continually logged at $\approx$ 16Hz throughout the study. The participants underwent a short training session to train their logistic function parameters, during which they were asked to perform example within-place and between-place activities, e.g. sitting at a desk or walking, while carrying the device in a pocket or a bag. Each training session lasted 30 seconds per activity type, and the parameters were trained using regularised logistic regression with stochastic gradient descent. Full instructions that were given to the participants are listed in Appendix B.

The participants were then asked to undergo their previously identified place sequences and perform their previously identified example activities in each place whilst carrying the mobile device in exactly the same manner as they would at and between each place. A researcher shadowed each participant and recorded the timestamp for the transition points into and out of each place – notified orally to the researcher by the participant themselves. Due to the difficulties involved in collecting such fine-grained data over an extended period of time, the participants were asked to perform shortened versions of activities in each place, e.g. "working at desk", which would typically last for 1–2 hours, was shortened to 5–10 minutes. The transitions between places were not shortened.

Analysis

Upon completion of the hybrid study, we analysed the data in order to observe the effects of each factor on the transition detection process. The factors are w, the weighting method for the moving average filter; and τ, the time window for the moving average filter. Although the data was *logged* at 16Hz, we *sample* from it at 1Hz, so the difference between timesteps k is constant at 1 second. For the participant-specific logistic regression parameters, θ_i, we used each participant's training data to find the maximum likelihood parameters – $\hat{\boldsymbol{\theta}}_i$ – for that participant i. Once found, $\boldsymbol{\theta}_i$ was held fixed at $\hat{\boldsymbol{\theta}}_i$ for each participant i during analysis.

We chose three moving average weighting methods, w, to analyse (see Figure 4.5):(i) the simple moving average (SMA), where all classifier outputs in time window τ are given equal weight; (ii) the weighted moving average (WMA), where the classifier outputs are weighted linearly over τ (with more weight given to recent motion); and (iii) the exponential moving average (EMA), where the classifier outputs are weighted exponentially over τ as follows, $\frac{2}{T_n+1}, T_n \leq \tau$, where T_n is the time-lag between the current timestep k and timestep n. As a benchmark, we also tested the process with no moving average. Finally, we evaluated 9 time windows τ at intervals of 5–10 seconds over 5–60 seconds.

Performance Measure

To measure the performance of each design, we use the precision, recall and F1 scores which account for true positive (tp), false positive (fp) and false negative (fn) classifications. Their descriptors are as follows:

- A true positive (tp) occurs when the process classifies a correct transition point according to observed data, i.e. a place entrance or exit transition. This must be made within an acceptable time window from the observed transition point, accounting for moving average lag τ.

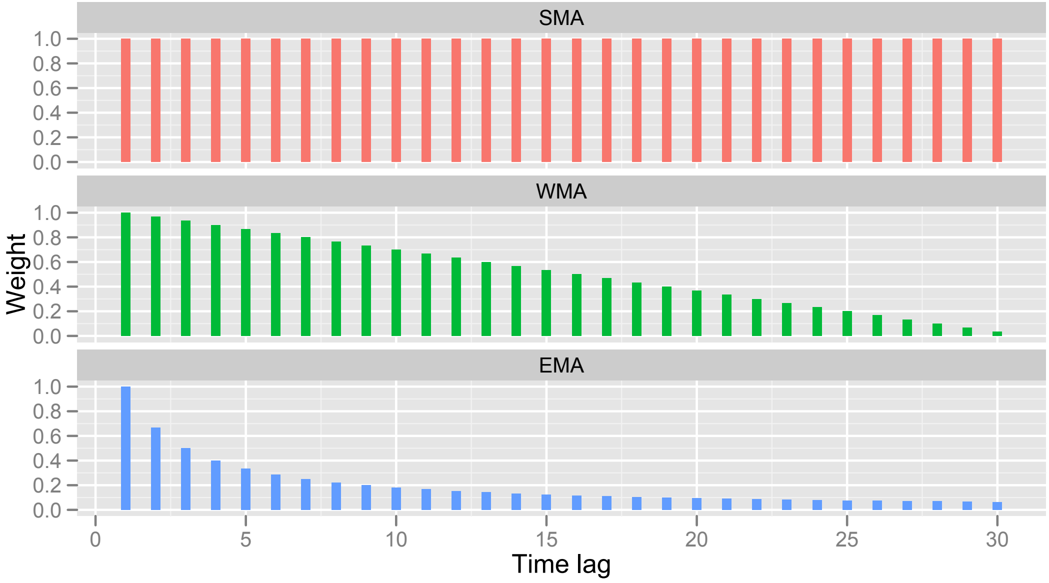


Figure 4.5: *Illustration of moving average weighting methods w as a function of time lag for $\tau = 30s$*

- A false positive (fp) occurs when the process classifies an incorrect transition point according to observed data, i.e. classifying a transition outside a viable observed transition.

- A false negative (fn) occurs when the process fails to classify a transition point according to observed data, i.e. not classifying a transition at the time of a viable observed transition.

To account for small deviations between the researcher-recorded observations and the exact moment of place transition, an observed transition was considered viable for 5 seconds either side of its recorded timestamp.

4.4.2 Results

For the study, the environment for participants 1–6 was a university campus; 7–13 was an office; and for 14 it was a town centre. The median number of transitions for the 14 participants was 16. Common place labels included: “desk”; “café”; “canteen”; “lecture hall”; “car park”; “lab”; “gym” and “meeting”. Common within-place activities included: “working”, “eating”, “reading” and “relaxing”. Common between-place activities included: “walking” (all participants); “cycling” (participant 4); and “driving” (participant 14). Figure 4.6 shows a graph representation of Participant 7’s place transition sequence, with each edge corresponding to two transition points (en-

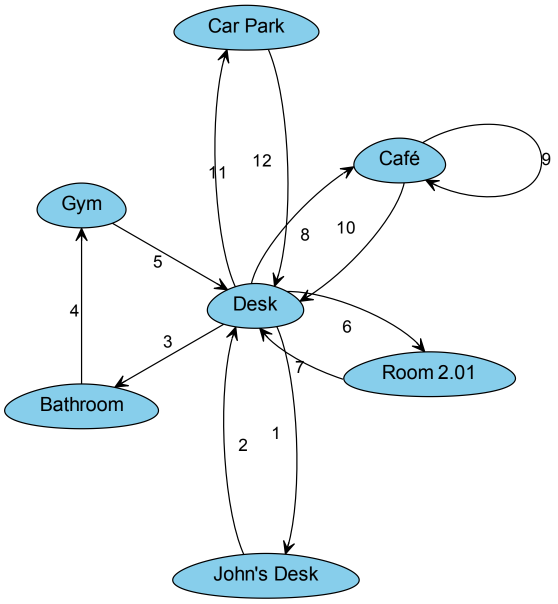


Figure 4.6: *Graph of the place transition sequence for Participant 7. Edge labels represent the transition ordering, and each edge represents an exit and entrance transition point.*

trance and exit). Figure 4.7 shows the transition point distribution over each of the 14 participants.

A two-way, within subject analysis of variance (ANOVA) over the factors w and τ shows that neither have a significant effect on each participant's true positive tp and false negative fn count ($F_{2,26} = 1.483$, $p = 0.25$ and $F_{8,104} = 1.746$, $p = 0.10$ in both cases, respectively), but there is a significant interaction effect ($F_{16,208} = 7.463$, $p < 0.01$). The same ANOVA shows that both w and τ have a significant effect on each participant's false positive fp count ($F_{2,26} = 21.25$, $p < 0.01$ and $F_{8,104} = 30.05$, $p < 0.01$ respectively) as well as a significant interaction effect ($F_{16,208} = 17.87$, $p < 0.01$).

By encoding the tp, fp and fn counts into between-participant comparable statistics – precision, recall and F1 score – we can perform a *post hoc* analysis on the effects of the factors upon performance. As the distributions of these statistics over the participants are unknown (and not well modelled using a normal distribution), we use a non-parametric bootstrapping method with 1000 replicates to estimate the mean and percentile confidence intervals of each statistic over the participants. Figure 4.8 shows the mean precision, recall and F1 scores for each w over τ. There is a significant observed improvement in precision and F1 score from no moving average by all levels of w for $\tau > 5$s ($p < 0.05$). There is a significant observed improvement in F1 score ($p < 0.05$) at $\tau = 30$s from $\tau < 15$s for the SMA; at $\tau = 50$s from $\tau < 25$s for the

WMA; and at $\tau = 60$s from $\tau < 20$s for the EMA.

Figure 4.9 shows a more detailed overview of the F1 score distribution for the participants, partitioned by w, at $\tau = 30$s.

4.4.3 Discussion

Here we discuss how the results address RQ 3.1, as well as the observations, limitations and implications of our approach and results.

RQ 3.1: To what extent can we infer significant transitions between users' meaningful places using mobile device motion?

The results suggest that the majority of users' self-defined significant place transitions can be inferred in real time using mobile device motion data passed through a logistic function and moving average window. The results further show that moving average time window τ and weighting method w (and their interaction) have a significant impact on inference performance. There is weak evidence to suggest that a simple moving average window is better than a linearly or exponentially weighted one over time windows ≤ 60s, and that the (simple) moving average window should be ≥ 15s for significantly improved performance.

Observations

The significant improvement from the absent case by all moving average types w in Figures 4.8 and 4.9 suggests that smoothing transient motion and requiring *sustained motion over time* is an effective method of detecting place transitions.

Interestingly, there is a peak in F1 score performance for the SMA at $\tau = 30$s (see

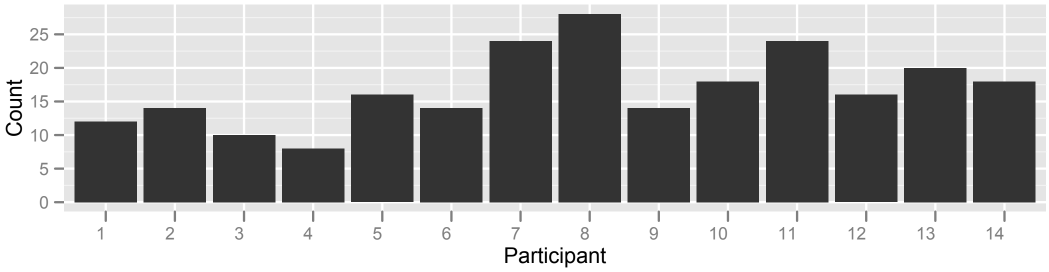
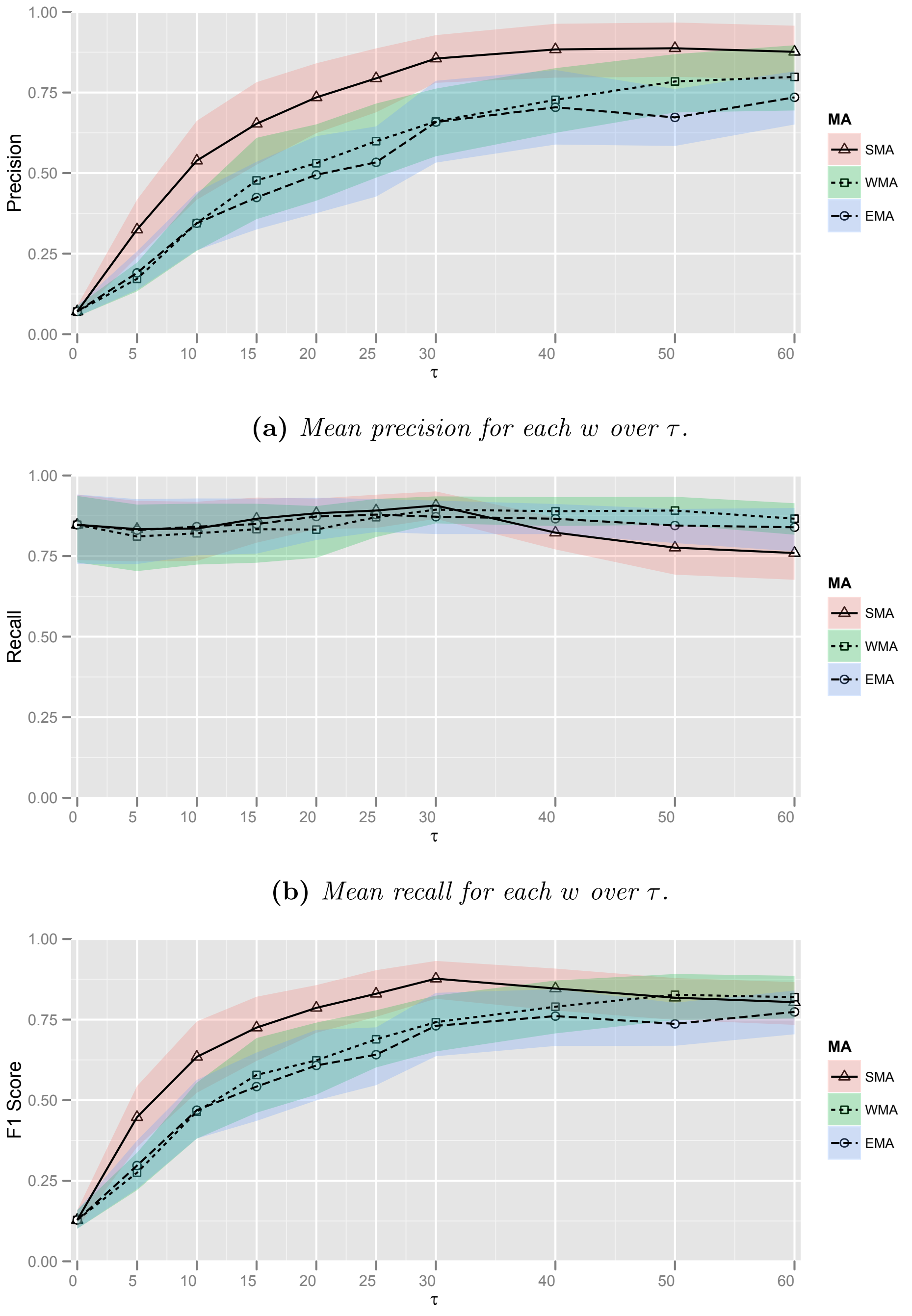


Figure 4.7: *Transition count distribution over the participants.*



(a) *Mean precision for each w over τ.*

(b) *Mean recall for each w over τ.*

(c) *Mean F1 score for each w over τ.*

Figure 4.8: *Plots showing the mean precision, recall and F1 score of transition inference performance for the MA weight type w over the time window τ.(.95 CIs shown.)*

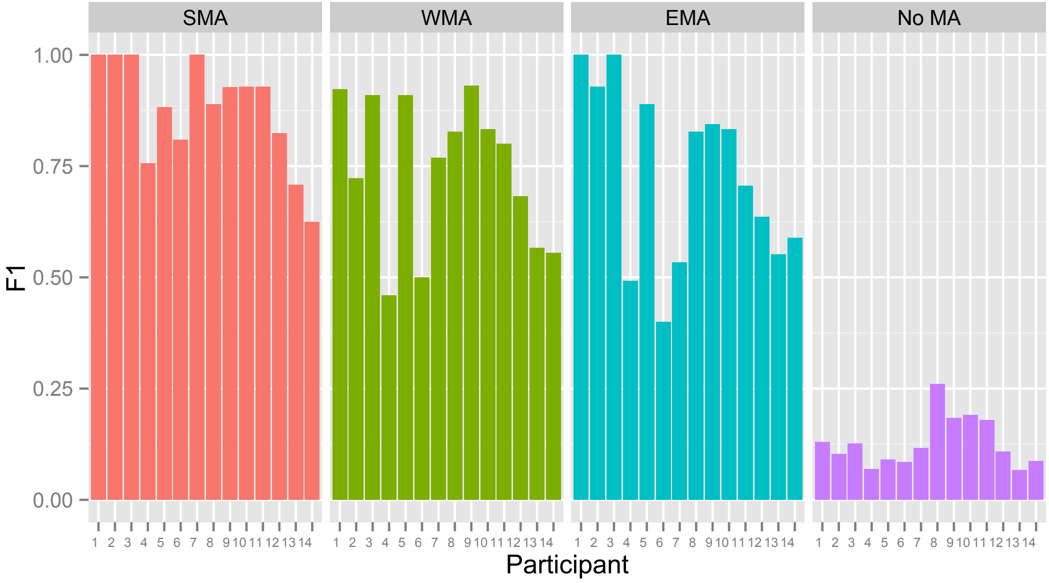


Figure 4.9: *F1 score, for each moving average factor, over the participants at* $\tau = 30s$

Figure 4.8). This is due to an increase in false negatives fn and consequent decrease in true positives tp causing a minor drop in recall as the time window τ increases beyond the shorter transitions for many participants, e.g. walking from a desk to a meeting which may take less than 30s. The peak is more apparent earlier for the SMA due to its unbiased weighting over the entire time window. The slight improvement using the SMA rather than the WMA or EMA shows that equal weighting of data in τ – rather than biasing toward recency – is likely to be the superior choice, not least because of the improvement in awareness latency (compare the approximately equal performance in Figure 4.8 of the SMA at $\tau = 20$s to the WMA and EMA at $\tau = 40$s).

Clearly the greatest improvement comes from reducing the per-participant false positive fp count. Aside from in-place idle motion (e.g. from the device in a pocket, or the user idly playing with it), these were generally caused by participants undergoing periods of 'start-stop' motion both within and between places, e.g. participant 12 using their device for a phone call; participant 6 moving within a large shop; participant 4 cycling; and participant 14 driving. A few false negatives fn were caused by participants leaving their device at their desk to travel to a nearby location, e.g. a printer (participant 7), or undergoing short transitions (with duration $< \tau$), e.g. stopping to talk to a colleague *en route* to another location (participant 13).

Notable observations from our hybrid study approach included the lack of cognitive overload for the participants. With the shadow monitoring them, some participants noted that they didn't have to "stop and think" (participant 13) about writing something down, or what they should be doing next, although some participants noted that – as well as the presence of the shadow – the time shortening during places felt a little artificial, even with them performing their natural activities in each. Another notable observation was the participants' willingness to undertake the study under the hybrid conditions: the majority said that – for privacy reasons – they would not undertake the study if the shadow (or other observation device, e.g. a camera) were to be present throughout their entire day, i.e. a full observational field study.

Limitations

One of the key limitations of the study were the environments. We were focused on 'local' environments – offices, campuses and a town centre – so we cannot easily generalise the performance from these results to multiple, more global, environments. Early indications of how the detection process deals with vehicular motion, i.e. participant 14, suggest that the stop-start nature of driving will impact on performance due to the fixed threshold of the trained classifier. However, using multiple fixed or adaptable logistic functions (e.g. one for each mode of travel) may alleviate these problems. Furthermore, data fusion with other sources of context data that suggest, for example, that the user is in a vehicle, e.g. in-car Bluetooth or GPS speed sensing, could improve performance in these situations, as could incorporating others' work into detecting transport types from mobile device motion, e.g. [191].

There are also limitations with the ecological validity of the hybrid field study. First, although the participants were asked to carry their mobile device in a naturalistic manner, e.g. in a pocket or bag, we could not capture entirely realistic idle motion profiles due to the shortening of the context periods. Furthermore, the performance of the participants' activities was necessarily artificial, i.e. they were enacted for the purpose of the study rather than to achieve a specific goal which could, in turn, affect the ecological validity of the captured motion data.

Implications

The output of this work results in a lightweight mobile service that can report genuine place transitions to any application that requires them. This has important implications for applications that focus on place recognition, e.g. [117], notification delivery systems, e.g. email or SMS, or in-situ user prompting. The results and feedback from using the hybrid study approach show that it (the approach) can be used to acquire useful results that could not otherwise be obtained reliably through laboratory or field studies.

4.4.4 High-level Implications and Limitations

RQ 3.1 asked the extent to which we can infer significant transitions between people's places using mobile device motion. What are the implications of our findings for RQ 3 – which asks about the extent to which significant changes in context can be inferred – and the limitations? The findings from this study suggest that mobile device motion and transitions between places form a good basis and trigger for observing changes in other context facets. Going back to our discussion about *when* the best time to sense might be given limited mobile device resources, using place transitions could bootstrap other processes for inferring changes in, for example, activity or intent. We have shown that, at least for the 'Where' facet, motion is a good indicator of certain context transitions, but where does this leave other facets?

Clearly, the biggest limitation of these findings is that of generalisation – particularly to the remaining context facets in the Five Ws. Although motion has been shown to be a useful indicator of activity transition [92] (What), little has been done to address subtler yet important transitions between intent, emotion or social network. These would be valuable to know; in the case of intent transition, this could dictate the entirety of the remaining facets' transitions, e.g. someone suddenly dropping in to a café for a drink en route to elsewhere as they unexpectedly saw some friends inside: the Who, What, Where, When and Why facets have all effectively transitioned as a result of this change in intent.

The next stage to better answering RQ 3 might be to repeat a scripted tour method with additional sensors in order to infer other context transitions beside place transitions. The broader the range of context transitions studied, the more knowledge is gained for researchers tackling this problem. In summary, our findings show that place transitions are possible to infer well from mobile device motion data, but that the study should be repeated for other sensors and context transitions in order to gain a broader picture of context transition inference in general.

4.5 Study: Inferring and Actively Learning Places

Now that we have seen how place transitions can be inferred through mobile device motion, we address RQ 4.1.

As related work in Section 4.3.1 showed, inferring and learning about people's meaningful places can be non-trivial. Determining the "Where" of places, i.e. their approximate spatial area, and the "Why", i.e. their meaning, are the key primary tasks for inference and learning. For the "Where", the state of the art inference approaches use RF fingerprinting, e.g. [41, 115] while geometric approaches, e.g. [11], are not as popular currently due to granularity issues with GPS sensing. This is a limitation of sensing technology however, not the inference approach. Location sensing in mobile devices is continually improving (see Section 2.4.4) and – at the time of writing – Google's Android operating system ships with a very accurate location sensing system, even indoors[3].

Whereas many geo-location services can identify the user's position and high-level, e.g. a street name, it is still difficult for a system to infer the personal meaning of people's places without being explicitly told. This is where why incorporating user place feedback into the system is an interesting prospect. The feedback – if successfully elicited from the user – has two key advantages: (i) places can be created directly from the user's labels and seeded from a location observation; and (ii) places can evolve over time as the user interactively trains the system through their mobile device using further location observations. This is particularly appealing for dealing with loosely defined places, e.g. a meeting 'area', and place descriptions that update over time, e.g. new areas of an office building or shopping mall. MIT's OIL system [173] uses a similar approach to mapping indoor wireless Voronoi regions. Although the approach is limited by bespoke hardware, it shows how learning from user feedback in real time can improve (in this case) wireless zone mapping.

One important challenge in any system that incorporates user feedback is the elicitation of the feedback from the user. In mobile context aware systems, prompting the user to intervene may be annoying and – depending on the user's context – they may not be aware of the notification or choose not respond to it. However, without feedback, context learning and future inference performance may be poor. A good approach will attempt to maximise automated inference and learning while minimising the amount of user feedback necessary for good performance.

In this study, we apply our general context inference and learning algorithms (Algorithms 4 and 5) to place awareness. We incorporate the work from Section 4.4 to infer transitions between places and use the transitions to bootstrap Algorithm 4. The problem of actually eliciting feedback from users is addressed in Chapter 5, but in this study we simulate feedback using the high-precision data collected from users. This allows us to implement Algorithm 5.

We first show how we implement the algorithms for place awareness, before detailing our study and simulation approach. The results show that, given expected user feedback, good place inference performance can be achieved as the system actively learns about users' meaningful places. We also compare three inference approaches and show how

[3] see `http://developer.android.com/reference/android/location/LocationManager.html` for more detail (Accessed 2012-11-07)

Algorithm 6 Place inference and active learning (executed upon transition into a new place)

```
Input: set of existing places $P$, their codebook vectors $C$ (indexed by place) and a confidence threshold $t$.
Output: the set of places ranked by probability $\hat{P}$, the inferred place $p \in P$ and confidence $c \in [0, 1]$, an updated set of places $P$ and their codebook vectors $C$
$\bar{\mathbf{x}} \leftarrow$ ObserveLocation                                ▷ Sensing location.
if $\bar{\mathbf{x}}$ is empty then
    return
end if
$\hat{P} \leftarrow$ InferPlaces($\bar{\mathbf{x}}$, $P$, $C$)                ▷ Place inference.
$p \leftarrow \hat{P}[1]$
$c \leftarrow$ MeasureConfidence($\bar{\mathbf{x}}$, $C[p]$)                 ▷ Inference confidence.
if $c < t$ then
    NotifyUser($\hat{P}$)
else
    UpdateCodebookVectors($\bar{\mathbf{x}}$, $C[p]$)                        ▷ Place learning.
end if
return
```

they affect inference performance. Finally we discuss how the results address RQ 4.1.

4.5.1 Approach

In this section we describe our approach, including design and implementation of the inference and learning algorithms as well as the user study.

System Design

The system is designed to run on a mobile device in real time. The algorithms are formally described in Algorithm 6 and Algorithm 7, which are instantiations for Algorithm 4 and Algorithm 5 respectively. The following subsections outline their implementation in greater detail. There are a number of alternative design choices that can be made at the inference stage; details of which will be outlined in the relevant subsection.

Place Representation

To address the problems of loosely defined places and place evolution over time, we represent each place as a set – or map – of 'codebook' vectors C in coordinate space. This enables two desirable properties: the ability to model places as probability density functions (PDFs); and allow them to be *malleable* over time using on-line vector quantisation, similar to training a self-organising map (SOM) [121]. Each codebook vector

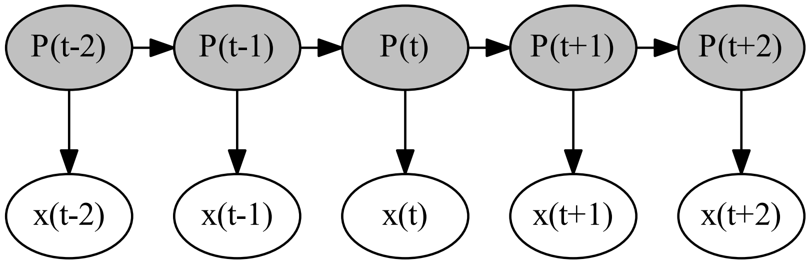


Figure 4.10: *Hidden Markov Model (HMM) of place inferences over time. There is a temporal dependency between each place variable.*

has a 'weight' associated with it represented by the number of location observations assigned to it – this is discussed in greater detail shortly.

Place Transition Inference

We use the place transition inference approach from Section 4.4, which uses mobile device motion to infer the moment of transition between meaningful places. This trigger initiates the execution of Algorithm 6.

Observing Location

Location sensors are used to report a set of location latitude-longitude coordinates and the *accuracy* of the estimate in m. This process is 'sensor-agnostic', i.e. it utilises whichever location sensors are operational on the device at the time, e.g. GPS or WiFi/Cell providers. If no location is obtained, the algorithm terminates and the user is notified that location data cannot be obtained. The user is notified of this for intelligibility and usability reasons: if the device simply cannot sense a location, e.g. there is no radio signal, it is better to inform the user of this rather than allow them to think that the system is performing poorly.

If a set of N location coordinates is successfully obtained, the weighted mean of the set is calculated:

$$\bar{\mathbf{x}} = \mathbf{L}^{\mathrm{T}}\mathbf{a} \tag{4.3}$$

Where $\mathbf{L}$ is a $N \times 2$ matrix of coordinates, and $\mathbf{a}$ is a column vector in $\mathbb{R}^N$ of normalised *inverse* accuracies. $\bar{\mathbf{x}}$ is then used as the location observation on line 3 of Algorithm 6.

Place Inference

For the place inference subroutine, we compare and contrast three alternative approaches:

1. **Hidden Markov Model (HMM)**:

   HMMs are probabilistic graphical models that are used for many applications, including voice recognition and natural language processing. Rabiner [185] provides an excellent introductory overview of HMM theory, construction and implementation. An HMM assumes the system can be in one of a number of *hidden* – or unobservable – states, and our observations of these hidden states are made through *observation* variables, whose fidelity of the hidden state may be affected by corrupting factors such as noise. HMMs are particularly useful for modelling processes that are temporal in nature, thus allowing the capture of temporal dependencies between successive states in a Markov process. HMMs are generally modelled under the assumption of the Markov property, i.e. the current hidden variable is independent of previous variables given its immediate (hidden) predecessor, and the current observation variable is independent of all other variables given its hidden counterpart.

   Figure 4.10 illustrates our application of HMMs to place inference, where the current place variable is dependent only on the preceding one according to the Markov property. Here, the place variable at timestep k, P_k, is a hidden variable that takes the value of one of the M_k existing places at time k and emits an observation that in our case is the location vector at timestep k: $\bar{\mathbf{x}}_k$:

   $$\mathrm{P}(P_k|\bar{\mathbf{x}}_k) = \eta \mathrm{P}(\bar{\mathbf{x}}_k|P_k)\mathrm{P}(P_k) \tag{4.4}$$

   Where η is a normalising constant described in Equation 4.6 below. Because of the Markov assumption, there is a dependency between the current place variable P_k and the previous place variable P_{k-1}. Using the chain rule of probability to re-factor the prior $\mathrm{P}(P_k)$ and by summing over the previous place variable, we have:

   $$\mathrm{P}(P_k|\bar{\mathbf{x}}_k) = \eta \mathrm{P}(\bar{\mathbf{x}}_k|P_k) \sum_{p_{k-1}} \mathrm{P}(P_k|p_{k-1})\mathrm{P}(p_{k-1}|\bar{\mathbf{x}}_{k-1}) \tag{4.5}$$

   Where conditional independence removes dependent paths between P_k and all location observations up to and including time $k-1$, as well as the dependent path between the current location observation $\bar{\mathbf{x}}$ and all previous place variables. Equation 4.5 contains the recursive component $\mathrm{P}(p_{k-1}|\bar{\mathbf{x}}_{k-1})$, which can be viewed as a ‘message’ passed forward through time [185]. The normalising constant η can therefore be written as:

$$\eta = \sum_{p_k} \mathrm{P}(\bar{\mathbf{x}}_k|p_k) \sum_{p_{k-1}} \mathrm{P}(p_k|p_{k-1})\mathrm{P}(p_{k-1}|\bar{\mathbf{x}}_{k-1}) \tag{4.6}$$

There are two parameters in these equations: the emission probability $\mathrm{P}(\bar{\mathbf{x}}_k|P_k)$, i.e. the probability of each place given the current location observation, and the transition probability $\mathrm{P}(P_k|P_{k-1})$, i.e. the probability of transitioning between places during successive timesteps. For the emission probability, we model each place as a multivariate Gaussian PDF in $\mathbb{R}^2$ and use this PDF as a likelihood function. The weighted mean and weighted covariance of the place's codebook vectors (weighted by observation count) are used as the maximum likelihood Gaussian PDF parameters. For the transition probabilities, we store an $M_k \times M_k$ transition count matrix $\mathbf{V}$ over the M_k places at each timestep and increment element v_{ij} upon observation of two successive confident inferences of places i and j respectively. $\mathbf{V}$ is initialised with a pseudo-observation count α over all $M_k \times M_k$ elements. $\mathbf{V}$ is initially normalised so that the row-wise transition probabilities sum to 1, and it is re-normalised on each count update to maintain this invariant.

If a new place is created, i.e. $M_k = M_{k-1} + 1$, a new row and column is added to $\mathbf{V}$ with pseudo-observation count α, before the rows of $\mathbf{V}$ are re-normalised to sum to 1.

2. **Bayesian Classifier (BC)**:

   Bayesian classification does not model direct dependencies between place transitions, and the posterior probability distribution at time k over the M_k existing places P given the location observation $\bar{\mathbf{x}}_k$ is calculated using Bayes' rule:

$$\mathrm{P}(P|\bar{\mathbf{x}}) = \frac{\mathrm{P}(\bar{\mathbf{x}}|P)\mathrm{P}(P)}{\mathrm{P}(\bar{\mathbf{x}})} \tag{4.7}$$

   Where the likelihood – $P(\bar{\mathbf{x}}|P)$ – is calculated as with the HMM by modelling each place as a multivariate Gaussian PDF in $\mathbb{R}^2$ and using it as the likelihood function. The weighted mean and weighted covariance of the place's codebook vectors are again used as the Gaussian PDF parameters. We use a multinomial distribution as the prior over P which is calculated using a vector $\mathbf{v}$ of confident observation counts for each place, smoothed using a pseudo-count α, over the existing M_k places.

   If a new place is created, i.e. $M_k = M_{k-1} + 1$, the dimension of $\mathbf{v}$ is incremented and set to pseudo-observation count α.

3. **Nearest Neighbour (NN)** The nearest neighbour classifier simply computes the Euclidean distance from the codebook mean of each place in P to the lo-

Algorithm 7 User feedback algorithm for active place learning (executed upon user response)

```
Input: place $p_b$ classified pre-feedback, its confidence measure $c_b$, confidence threshold $t$, place $p_a$ classified post-feedback, $\bar{\mathbf{x}}$ the location observation for the place, $P$ the set of existing places and their codebook vectors $C$
Output: the classified place $p \in P$ with confidence $c$, an updated set of places $P$ and their codebook vectors $C$
if $p_a \neq p_b$ and $c_b > t$ then    ▷ Undo learning if incorrect inference was confident.
    ReverseUpdateCodebookVectors($\bar{\mathbf{x}}$, $C[p_b]$)
end if
if $p_a \in P$ then
    UpdateCodebookVectors($\bar{\mathbf{x}}$, $C[p_a]$)    ▷ Execute learning.
else
    $C_a \leftarrow$ GenerateCodebookVectors($\bar{\mathbf{x}}$)    ▷ New place added.
    add $C_a$ to $C$
    add $p_a$ to $P$
end if
$p \leftarrow p_a$
$c \leftarrow 1$
return
```

cation observation $\bar{\mathbf{x}}$. The location observation is then classified as the nearest neighbouring place.

The places are then ranked by probability (if applicable) or Euclidean distance from $\bar{\mathbf{x}}$ to give a ranked set $\hat{P}$, and the top-ranked place is given a confidence measure. For this we implement a one-sample Hotelling's T^2 test that compares the set of C codebook vectors to the observation $\bar{\mathbf{x}}$ and uses the p-value as the confidence measure.

The inferred place is then assigned to be the top-ranked place unless the user intervenes. If the confidence of the top-ranked place is less than a certain threshold t, a request for feedback is sent to the user and – regardless of notification – they are presented with the ranked list $\hat{P}$ for feedback.

If the inference is confident however, the user is not notified (though they can intervene without notification if they wish), and the system updates the top-ranked place's codebook vectors autonomously (see the following section).

Active Learning

User feedback is one of two possible actions:

- **Creation**: where the user creates a new place from a meaningful label and the current location observation $\bar{\mathbf{x}}$.

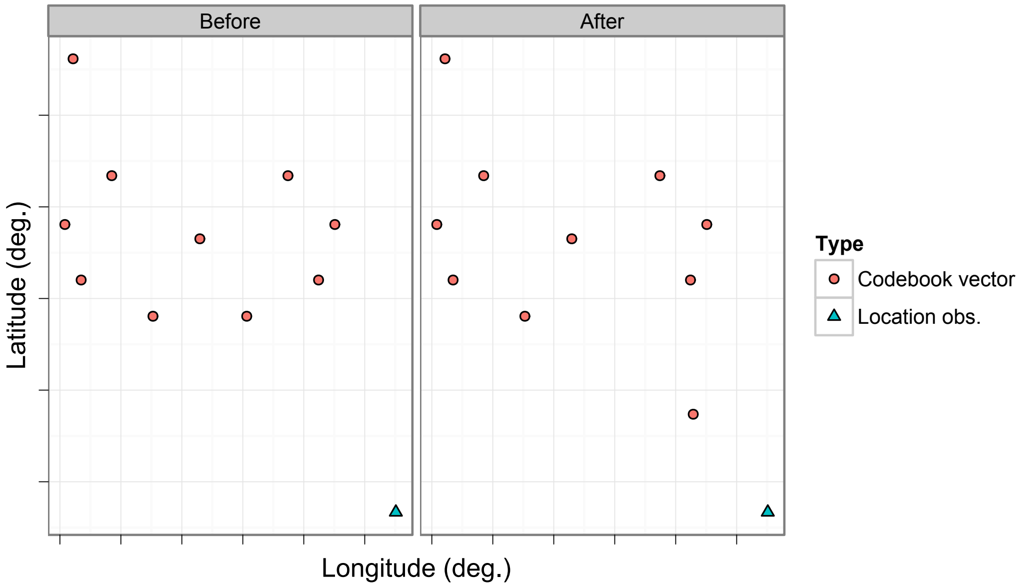


Figure 4.11: *Before and after updating a place's codebook vectors: the nearest codebook vector to the location observation vector is pulled towards the observation.*

- **Selection**: where the user informs the system of the correct place from the ranked set of places $\hat{P}$. This can be a confirmation of a correct inference (if the inference is not confident.)

A user can create a new place using a label of their choice. On doing so, a new set of codebook vectors are generated and the place is added to the set of places P. The generation process uses the location observation as a seed codebook vector and creates the remaining set of codebook vectors in a circular array around the centre with radius r.

Algorithm 7 is executed upon the user feedback action.

The learning process uses a modified, online version of the k-means algorithm [64] which updates places using their codebook vectors C and the location observation $\bar{\mathbf{x}}$.

$$\mathbf{c}_n := \mathbf{c}_n + \frac{1}{N_n}(\bar{\mathbf{x}} - \mathbf{c}_n) \tag{4.8}$$

Here, $\mathbf{c}_n$ is the Euclidean distance nearest-neighbour codebook vector to $\bar{\mathbf{x}}$, and N_n is the observation count for codebook vector $\mathbf{c}_n$, i.e. the number of previous observations associated with $\mathbf{c}_n$. The codebook vectors are therefore – upon observation – 'pulled' towards the location observations as in Figure 4.11. The observation count N_n enables convergence, so that the codebook vectors C for each place form a self-organising map.

The process can also be reversed if the user corrects a previously confident place inference (line 4, Algorithm 7). This is done by subtracting – or 'pushing along' – the difference vector in Equation 4.8 rather than adding it.

Study Design

Here we describe our user study where we analyse system performance using real-world data and simulated user responses to notifications. Our aim was to capture the expected output of the place inference and learning algorithms given typical user feedback behaviour. Unfortunately, this is practically infeasible to undertake in a field study: we can model users' place visits as a series – or chain – of transitions which, even with a simple binary feedback variable at each place in the chain, results in 2^n possible chain outcomes (where n is the number of place visits in the chain). We can, however, simulate *expected* outcomes by sampling from a feedback probability distribution at each step in the chain.

Data Collection
To collect the real-world data, we recruited 6 participants (all male, mean age 32), 3 of whom are office workers (IDs 2, 3 and 5) and the rest university students. Although 6 participants is a small sample, we were somewhat constrained by the resources involved in capturing reasonably long-term fine-grained observations of participants' place sequences in real time. This is a slighty larger sample size than those used in previous studies that undertake similar fine-grained observation, e.g. [90, 117]. Moreover, as the onus of this study is on simulated outcomes, we can artificially generate a larger dataset from the seed observations of 6 participants. In a pre-study interview, we asked them to describe their typical working week (5 days) as a series of transitions between self-defined meaningful personal places – along with typical activities within each place – partitioned between the start and end of each day. As before, we used the Dourish and Harrison example of 'place' vs 'space' [86], and examples from Barkhuus *et al.* [17] to explain the difference between place and location to the participants. Following this, we equipped them with an Android 2.3 Nexus S device running the place transition detection system in [144]. At each transition *into* a place, the device attempted to capture 10 location samples with accuracy measures from its available location sensors: either GPS or Android's network-based provider depending on which was available. Samples with an accuracy of $>$ 100m were rejected – which is a standard rejection threshold used in similar studies, e.g. [115, 117], and a sampling timeout period was set to 60s.

Each participant was then instructed to undergo a scripted tour of their places in their self-described order of transition whilst carrying the device 'naturally' throughout, i.e.

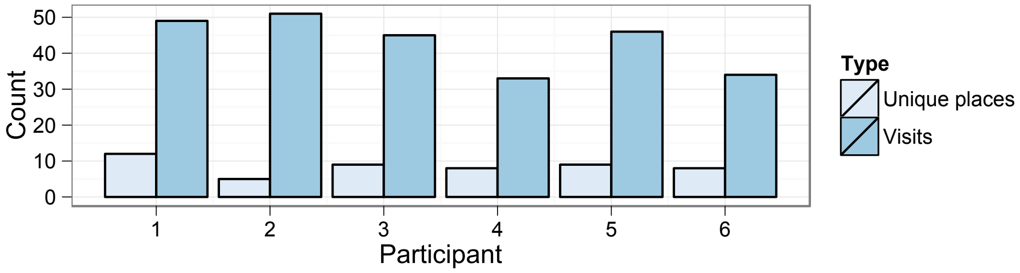


Figure 4.12: *The number of unique meaningful places and total visits, i.e. place sequence length, over the participants*

as they would carry their own device within and between each place. A researcher shadowed each participant and recorded what actually happened whilst the participant transitioned between places and performed their activities within each place. As recording a week's worth of observations in this manner is very time consuming, the participants were asked – if possible – to shorten the duration of time spent within each place to no less than 5 minutes. The transitions between places were not shortened.

Within each place, the researcher asked each participant to rate how likely – 'high' or 'low' – they would respond to an audio, visual or haptic notification from their device given their current context. A more detailed account of the instructions given to the participants in this study can be found in Appendix B.

Simulation Design

Once the device location traces, participant observations and notification response ratings were captured, we designed a simulation to measure expected system performance given the large number of possible feedback outcomes for each participant's place transition chain. A single run of the simulation steps through each participant's place transition chain and – at each recorded transition – executes Algorithm 6. If a notification is raised, the decision as to whether the user responds is sampled from a Bernoulli distribution with parameter p_r set by the participant's recorded 'high' or 'low' notification response rating. If true, the simulation executes Algorithm 7.

As we are comparing 3 classification methods – the HMM, Bayesian classifier (BC) and nearest-neighbour (NN) classifier – we have 3 design choices for the place inference subsystem. Thus, each design is tested once on each transition chain in the simulation.

Performance Measure

We measure performance using the standard precision, recall and the F1 scores which are calculated using the following metrics:

- A **true positive** occurs when the inferred place at each step in the chain matches observation at the time of inference.
- A **false positive** occurs when the inferred place does not match observation at the time of classification.
- A **false negative** occurs when the system fails to infer a place as indicated by observation.

User context is important when considering user feedback behaviour: the user may not respond to requests for feedback, e.g. they do not notice them, or choose not to respond to them; or – if they do respond – they may do so at any time within their current place. Thus, we measure the performance both before and after simulated feedback responses (if any) within each place. This has real-world implications: if inferences differ pre and post user feedback, services and applications that rely on real time place inference may be affected.

- **Pre-feedback** performance measures the F1 classification performance *before* the user intervenes – if at all – within each place.
- **Post-feedback** performance measures the F1 classification performance *after* the user intervenes – if at all – within each place.

We measure user feedback by the number of place creations and selections as fractions of the total number of classifications. Moreover, we measure the level of system automation as the complement of the total feedback fraction.

4.5.2 Results

Class. Method	Precision (pre)	Precision (post)	Automation	Sum
BC	0.70400	0.80738	0.78224	2.29362
HMM	0.70398	0.80737	0.78225	2.29359
NN	0.56944	0.77273	0.77212	2.11430

Table 4.1: *Ranked table of designs, sorted by the sum of pre-feedback precision, post-feedback precision and automation.*

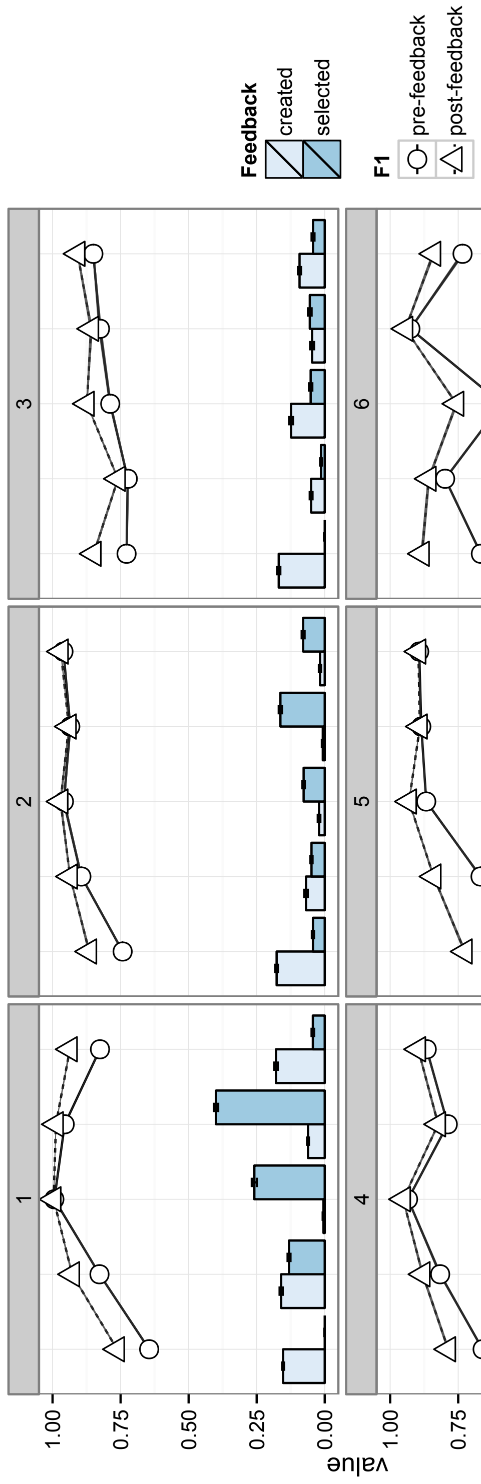


Figure 4.13: *Pre and post feedback $F1$ performance – in addition to user feedback actions as a percentage of total classifications – for each participant over the 5-day scripted tour for the BC design.*

For these results, the Bernoulli distribution parameter p_r was set to 0.25 for each participant's 'low' response rating and 0.75 for each 'high' response rating. These were chosen as they are the quantiles that equally bias the postive (< 0.5) and negative (≥ 0.5) sides of the Bernoulli distribution. This choice may affect the simulation output, so further work could include analysising the validity of thie parameters. 1000 simulations were run for each of the 6 participants' datasets and the expected F1 performance is aggregated within each participant over these simulation runs. Each run is seeded by the initial place in the participant's transition chain. We used 10 codebook vectors per place with radius r of 10m (using the assumption that places can 'grow', such that we can better catpture smaller places), a classification confidence threshold t of 0.05 – which is the p-value of statistical significance for the Hotelling's T^2 test – and a pseudo-count vector α of 1 over all places to avoid division by zero in the probablistic classifiers. Again, an agenda for future work could explore how significantly the variation of these parameters affects performance, if at all.

Figure 4.12 shows the number of unique places chosen by each participant along with the number of transitions in their chain, i.e. the total number of place visits over the 5-day scripted tour. Most places were indoors, and common labels used included "desk", "canteen", "bus stop" and "café".

Location sample accuracy statistics – mean (and sd) in m for each participant in order – are: 23 (6); 21 (3); 21 (5); 24 (7); 21 (5) and 27 (11).

Inference Performance Results

A two-way paired Student's t-test between each classification design pair over the performance metrics shows that: there is no significant effect on pre-feedback true positives ($t_5 = 0.063, p = 0.95$), pre-feedback false positives ($t_5 = -0.063, p = 0.95$), post-feedback true positives ($t_5 = 0.063, p = 0.95$) or post-feedback false positives ($t_5 = -0.063, p = 0.95$) when comparing the Bayesian classifier (BC) to the HMM.

There are significantly more pre-feedback true positives and significantly fewer pre-feedback false positives when comparing the BC and the HMM with NN ($t_5 = 3.934; p < 0.05$ and $t_5 = -3.934, p < 0.05$ respectively for BC; $t_5 = 3.933, p < 0.05$ and $t_5 = 3.933, p < 0.05$ respectively for HMM). The same comparison for post-feedback true positives and false positives reveals no significant effects ($t_5 = 2.463, p = 0.06$ and $t_5 = -2.463, p = 0.06$ respectively for BC; $t_5 = 2.462, p < 0.06$ and $t_5 = -2.462, p = 0.06$ respectively for HMM).

Table 4.1 ranks the designs based upon the sum of pre-feedback precision, post-feedback precision and automation. (Recall was invariant across the designs.)

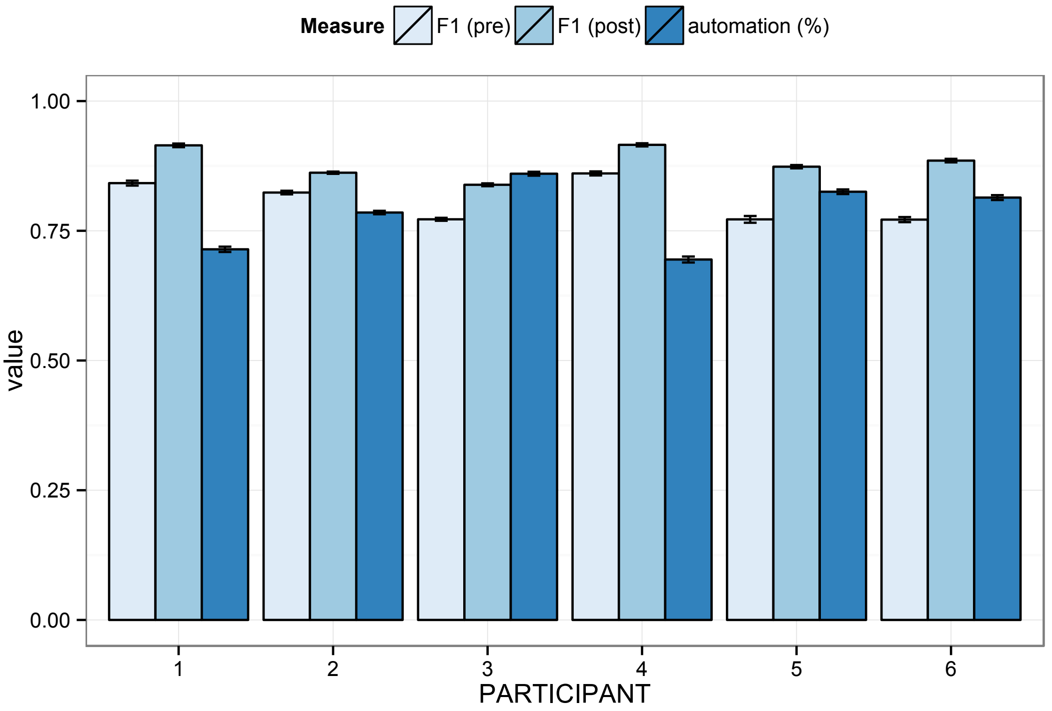


Figure 4.14: *Summary statistics for the best design in Table 4.1 (BC) over the participants for the 5-day scripted tour. Automation measures the fraction of classifications made without user feedback. (.95 CIs shown.)*

Further Results

Figure 4.13 shows the pre-feedback and post-feedback F1 scores for each participant over the 5-day scripted tour using the BC (the top ranked method in Table 4.1); along with the measures of user feedback. Figure 4.14 shows the aggregate performance over the 5 days, in addition to the automation measure, i.e. the fraction of automated classifications (made without user feedback).

The overall mean scores across the participants are 0.81 (pre-feedback F1 score), 0.88 (post-feedback F1 score) and 0.78 (automation %). User feedback shows a significant improvement in F1 score for each participant ($p < 0.01$ in each case, non-parametric bootstrap; 10000 replicates; $N = 1000$).

4.5.3 Discussion

Here we discuss how our results from the study address RQ 4.1, as well as further observations and limitations of our approach.

RQ 4.1: To what extent can we infer and actively learn about users' meaningful places using mobile devices?

The results suggest that good place inference performance can be achieved using active learning with a small amount of user feedback using our approach. This is shown specifically by the high inference performance and automation measures across the participants in Figure 4.14. Moreover, as Figure 4.13 shows, performance remains reasonably consistent as more places are created and visited by the participants over their working week. Not only are the places inferred well, but the user feedback aspect means that labels can be captured from the user *in situ*; which integrates place meaning into the learning process.

Which Inference Approach is Best?

As the findings suggest, performance differences between the BC and the HMM inference approaches are negligible, but both appear to be a better choice than NN. The significant improvement by the HMM and BC over NN is likely due to the probabilistic approaches (BC and HMM) utilising the weighted covariance of each place's the codebook vectors in the inference process. NN simply uses the weighted mean of each place's codebook vectors to calculate Euclidean distance, which suggests the weighted covariance component of the learning process is an important contributor to performance.

The key difference between the BC and the HMM lies in the direct temporal dependency between places in the HMM. With the dataset being reasonably small in both sample size and duration, the benefit of the HMM – if any – is not significantly realised due to the large amount of data required for realistic modelling of transition probabilities. Based on our findings and the desire for parsimony, i.e. the negligible performance difference between the BC and the HMM, the BC appears to be a better design choice for place inference. At each timestep k, and for M_k places at k, the HMM requires the update and storage of the $\mathbb{R}^{M_k \times M_k}$ matrix $\mathbf{V}$ compared to the BC's update and storage of the $\mathbb{R}^{M_k}$ vector $\mathbf{v}$.

The findings would suggest the BC to be the better choice, but more data over a longer period of time is required to fully answer this question, i.e. whether or not the HMM can significantly improve performance through the incorporation of temporal dependency between places.

Further Observations

For the BC design, the 'dip' on day 3 for participant 6 is caused by a place generation from a noisy initial location observation, exacerbated by repeated visits with good quality location data and subsequent incorrect inferences. This raises an important data quality and user interface issue: user interfaces for feedback should provide a function to 'reset' or 'recalibrate' places if seeded from inaccurate location data.

Later in the week, the pre-feedback performance begins to converge on the post-feedback performance (Figure 4.13). This is due to fewer place creations by users coupled with increased automated classification performance from place learning.

Even though the majority of the participants' places are indoors, the location sensing capabilities of the Android devices are surprisingly good; as shown by the location accuracy summary statistics, even with a rejection threshold of 100m. (Raw accuracy data from the field deployment in Chapter 5 can be found in Section C.4 of Appendix C.) The chief cause of false negatives, however, was location sample accuracy exceeding this threshold – the system is designed to make no classification rather than one with noisy data.

The key causes of false positives were false transition detection from the motion triggering system and incorrect classifications made by the classifier.

Overall, we have shown that capturing, recognising and learning meaningful personal places in real time on mobile devices is feasible with well-timed user prompts and a small amount of user feedback. Moreover, we have shown that the user can train the device to recognise places from repeated location observations over time using a form of active learning. This is a step beyond automated place recognition approaches, as it allows for content to be captured in addition to place 'malleability' and evolution over time as more location samples are observed.

The Limitations of Simulation

Although these early findings are both interesting and promising, there are some limitations to them. First – although simulation allows us to analyse many user feedback outcomes from a single dataset – it does not provide the implementation and behavioural insights that a long-term field study can offer. Second – although highly precise and fine-grained – the observed datasets are reasonably short and obtained from a small number of participants. Therefore further study is needed to fully investigate the long-term performance and behaviour of the system.

Another key issue is that of modelling user response to notifications. For these simulations, we have assumed that the user is a 'perfect oracle', i.e. they give the correct answer when responding to notifications. Users are of course prone to error [198], e.g. spelling errors, and this could have an impact on performance. Moreover, we did not model user feedback *without* notification, i.e. users providing feedback of their own accord without notification from the system. Furthermore, we have not specified what type of interface the user would be using to intervene with the system – merely that they create and select places; therefore the type of interface used in deployment is also an important aspect for design and implementation.

There is also the issue of complexity: modelling user feedback behaviour using a Bernoulli variable is simplifying what is, in reality, a complex process. Whether a user provides feedback or not can depend on many factors, including external stimuli in the user's environment and – of course – upon the user's context. Modelling the feedback process using this 'black box' method does at least allow us to observe system behaviour in response to varying feedback inputs, but it also hides latent variables that may otherwise be non-trivial to both observe and model.

Finally, there is the issue of arbitrariness. During our simulation analysis, we used a Bernoulli distribution with parameter p_r to sample from for our feedback variable, and we used a 'high' and 'low' measure by asking the participants to rate how likely they would respond to an feedback prompt. Of course the participants can give an estimate of this, but their response may not consider other complexities and stimuli that may otherwise be a factor for their true response behaviour.

4.5.4 High-level Implications and Limitations

RQ 4.1 asks the extent to which we can infer and actively learn about people's places through their mobile devices. The high-level RQ 4 asks this but for context in general. Our findings from this study show that active learning is a viable and useful approach to infer the 'Where' and, to some extent, the 'Why' facets of context. By capturing user feedback *in situ* through active learning prompts, we can not only improve inference accuracy but also capture richer feedback from users. In our case, it was place names, but this could extend to other forms of context including meaning or feeling. Similar to Experience Sampling [50], the potential to capture rich data at the relevant time is improved, but perhaps the most important implication is the ability to 'offload' much of the work onto the intelligence. Given enough active learning, the process of experience sampling could be – for the majority of the time – automated, with little burden to the user.

How might we infer and actively learn about other context facets? Others, particularly the 'What', are likely to be trickier. Users who interact with their device to provide feedback on their activity are, through the act of providing feedback, altering their activity. This could be alleviated somewhat by introducing latency between inference and the prompts for feedback, but this itself is a further inference and prediction problem. This example serves to illustrate the potential difficulties of inference and active learning of other facets, but further study would be beneficial to further answering RQ 4.

What our findings have further shown, however, is the potential for capturing richer data such as emotion and intent. In Chapter 3 we discussed the difficulty of sensing such context data due to its inherently complex nature. The best 'sensor' of this data is the user themselves, and through active learning we can begin to transfer this data from the user to the intelligence in a way that modern sensing cannot achieve.

The main limitations of this work are, again, generalisation. As we have just discussed, there are other facets for which it would be interesting to study the potential for active learning, and whether we would see similar inference performance improvements with them. Furthermore, other inference approaches and active learning strategies could be compared in order to build a fuller picture for RQ 4.

4.5.5 Towards Deployment in the Field

Following our discussion on the limitations of simulation, it would be prudent to deploy the place awareness system in the field in order to observe inference performance and users' feedback behaviour when in their natural environments. Simulation has allowed us to systematically analyse different inference approaches, but it cannot capture the complexity of the field environment. A field study – although limited in the amount of control and in-depth analysis that we can perform – will allow us to observe the feedback behaviour and system outputs in a more realistic setting.

4.6 General Discussion

In this section, we reflect on how the work in this chapter has addressed the higher level research questions for the intelligence component of interactive intelligence. We also discuss further generalisations of the work.

4.6.1 RQ 3: To what extent can we infer significant changes in context using mobile devices?

By considering the dynamics of context as a FSM that transitions between states of context over time, we identified the transitions as the points of significant change. Through the use case of place awareness, we have shown how transitions between users' meaningful places can be inferred using mobile device motion and how performance varies with the parameters of this approach. Although we have only shown this for place transitions, motion can also be used to infer significant context changes in other cases, e.g. activity transitions [92, 214]. Possible avenues for further work relating to the inference of context change include: extracting features from other context sensors, e.g. the microphone or the camera, and evaluating how effectively they can be used to infer context changes.

4.6.2 RQ 4: To what extent can we infer and actively learn about context using mobile devices?

For this question, we developed the general context inference and active learning algorithms shown in Algorithm 4 and Algorithm 5. Algorithm 4 is designed to be triggered at the moment of context change (linking back to RQ 3) and Algorithm 5 when the user provides feedback to the system. The idea is that the algorithms will perform context inference and learning in real time; learning as it goes along any only querying the user for feedback when necessary.

We applied the algorithms to the place awareness use case, using the place transition inference approach developed to address RQ 3. Results from a user study and simulation showed that good place inference performance can be achieved with expected user feedback behaviour. Furthermore, we showed how performance can vary according to the inference method chosen; with probabilistic inference methods (HMM and BC) outperforming distance-based methods (NN).

Although using simulation limits the impact of our findings, they do suggest that the algorithms are viable and that we can infer and *actively* learn users' context through their mobile devices. Along with questions surrounding how we might elicit user feedback in response to active learning queries, there is a clear need for study in the field to further validate the findings of the work in this chapter.

Other paths for future work include: the application of Algorithms 4 and 5 to other context awareness cases, e.g. activity, user identity or emotion inference and learning; evaluating the energy-accuracy trade-offs of the context transition triggered algorithm *vs*, e.g. periodic or random triggers; and extending the single user model to multiple users, e.g. inferring and learning meaningful places for groups of people.

4.6.3 Generalisations

Our approach and findings from this chapter could be generalised to other use cases beyond place awareness. For example, context transitions could be used for raising notifications and user prompts in User Experience studies, e.g. as part of an Experience Sampling Method (ESM) [51]. By providing users with prompts at the critical time of context state change, responses could be more useful than if the users were prompted periodically or at random – a conjecture supported by the activity transition work of Ho and Intille [92]. Context transitions could also bootstrap other useful processes, such as: data synchronisation on mobile devices (email or file synchronisation); message delivery notifications; and location based updates that are useful to, for example, on-device map applications.

Our context inference and active learning approach could be generalised to other context awareness use cases, e.g. activity recognition. Active learning, and our implementation using SOMs in the place awareness use case, is not restricted to $\mathbb{R}^2$ location space. It can support multiple feature spaces in which unsupervised clustering techniques can be applied, e.g. activity clustering based on device motion and orientation features.

4.6.4 Implications and Limitations for Context

The primary context facets addressed by the work in this chapter are 'Where', 'When' and 'Why'. The key implication for a theoretical understanding of context relates to 'Why': the idea of transferring more complex knowledge from the user to the intelligence through active learning. Rather than attempting to 'sense' this knowledge directly, this gradual transfer is a new and potentially novel way of capturing other context such as human intent. At the cost of some user burden, mobile devices could be used as a mediator to elicit what technology currently cannot, and further study would be valuable in uncovering the extent to which this is possible.

We have also argued for the Harrison and Dourish interpretation of 'place' and 'space' being two different concepts [86]. By allowing users to supply place meaning through short labels, we can not only capture location, i.e. latitude-longitude or address, but also some meaning that, again, cannot otherwise be sensed directly. Our findings show that the somewhat abstract definition of place can be captured and updated through active learning, which can enable further personalisation of applications, mobile or otherwise.

We have further highlighted the importance of ‘When’ which, although is acknowledged as an important aspect of context [57], is rarely studied and utilised. Our work has shown that it is valuable, and capturing the moments of context change has value beyond simply triggering further inference and learning processes, e.g. notification delivery and resource management.

The main limitations of our work for context understanding lie in the choice of facets. We have not considered the ‘Who’ or ‘What’ (beyond movement in the transition study) facets, and therein lie avenues for further work. Applying active learning techniques to social network inference could be interesting, especially if using mobile devices to mediate the active learning. Perhaps allowing users to give feedback about who they are co-present with could enhance context inference and learning. There is also the possibility of sharing actively learned context models, e.g. a user who has trained an activity model through active learning could share it with friends, thereby removing much of the training burden associated with active learning.

In summary, our findings have furthered the understanding of the ‘Where’, ‘When’ and ‘Why’ facets of context, but further work studying the ‘Who’ and ‘What’ facets would allow for better understanding and application of context.

4.7 Conclusion and Chapter Summary

In this chapter, we have addressed RQ 3 and RQ 4 in the intelligence layer of our layer model. By modelling context as a FSM, we used the transitions between context states as event-based triggers for executing context inference and learning algorithms. We also designed two algorithms for context inference and active learning, in which the algorithm chooses the data it learns from based on a confidence (or uncertainty) measure. Bootstrapped by inferred context transitions, the algorithm can infer and learn about users’ context in real time.

We presented the use case of place awareness – a suitable case for study due to its relevance and opportunity for extension to existing work in the field. We directly addressed RQ 3 by showing that significant place transitions can be inferred through mobile device motion, and that varying the parameters of this approach has significant effects on performance. We then applied our inference and active learning algorithms to the place awareness case and showed that our approach is both viable and successful for place inference and learning.

In the next chapter, we will move up from the intelligence layer to focus on the interaction layer of our layer model. We will address the *interaction* component of interactive intelligence in mobile context aware systems; specifically how we might encourage and elicit user feedback in mobile IIS through a mobile user interface. We will use our findings from this chapter to design, implement and deploy the place awareness system in the field; where we will observe user behaviour and inference performance, as well as the efficacy of different feedback request approaches. We will also compare the simulation results from this chapter with results from the field.

Chapter 5

Interactive Intelligence: The Interaction

In the previous chapter, we addressed the *intelligence* aspect of interactive intelligence in mobile context aware systems, i.e. the AI aspect and how we might incorporate user feedback into the context inference and learning process. We presented an approach for real time context inference and active learning with user feedback, and used the concrete use case of place awareness to address RQ 3 and RQ 4. We presented two user studies, one of which simulated user feedback behaviour for the active learning component of our approach. Although the simulation study was performed using highly reliable observation data, it was still a simulation and perhaps not as informative as a more ecologically valid field study.

In this chapter we consider the *interaction* component of interactive intelligence in mobile context aware systems, i.e. the human aspect and how we might elicit and encourage user feedback for active context learning. We primarily address RQ 5 and RQ 6 – though we further address RQ 4 – by moving up to the user interaction layer in our layer model (highlighted in Figure 5.1).

To remind the reader, RQ 5 and RQ 6 are the following mid-level research questions derived in Chapter 2:

- **RQ 5**: How can we elicit context feedback from users in a mobile environment?
- **RQ 6**: How do users interact with an interactive intelligent mobile context aware system?

In Chapter 4, we used active learning as an approach for improving context inference through *in situ* user feedback. Eliciting such feedback from users in the field is itself

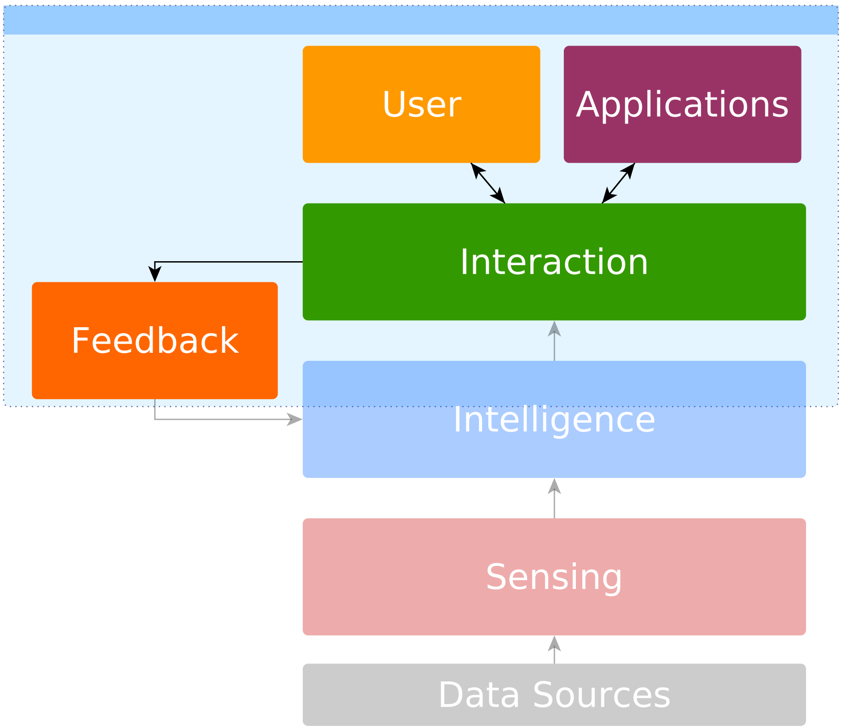


Figure 5.1: *Our layer model, highlighting the components that this chapter will be focusing on.*

a challenge due to unpredictable external stimuli and the diverse range of possible context states that users may experience in their daily lives. Therefore, to improve the validity of the work in this dissertation, we should consider how users might respond to feedback queries from the system, and how the feedback – if given – affects context inference in the field. This may reduce the reliability of our measurements, i.e. we cannot observe reality to such a high degree of precision as before, but we do gain knowledge of natural user interaction behaviour and the user experience from using an intelligent interactive mobile context aware system.

The main content of this chapter is the field deployment of an interactive intelligent mobile context aware system; namely the place awareness system developed in the previous chapter. The study is controlled such that we can compare different approaches to user feedback elicitation, and a *post hoc* performance test allows us to evaluate inference performance. We also present user feedback from the experience of using the interactive intelligent system.

We first outline the feedback and interaction requirements for the system. Following this, we report on the field deployment of our case study – the mobile place awareness system – in which we outline: the low-level research questions for the field study; directly relevant related work; the system design, particularly the design of the user interfaces for enabling user feedback; the design of a demonstrator presence and availability application; and the field study approach. Following this, we report the key results and findings from the field study – including place inference performance results from *post hoc* scripted tours – and discuss how the findings help to answer the low-level research questions.

The final part of this chapter reflects upon the high level implications and limitations of our work – specifically the implications for interactive intelligence in mobile context aware systems – and how they contribute to answering RQ 5 and RQ 6. We then review a set of potential application areas for the work developed in both this and the preceding chapter. Although the majority of our work is focused on enabling technologies for mobile context aware applications, it is important to address potential application areas.

5.1 Feedback and Interaction Requirements

As Stumpf *et al.* state, the incorporation of user feedback into machine learning systems is still an emerging practice [219, 220]. As such there are no standard requirements for user feedback and interaction in such systems.

Given this lack of standardisation, and in order to provide a basis for system design that relates directly to user needs, we develop and specify a set of abstract requirements for interactive intelligence in mobile context aware systems. These consist of two components: (i) feedback requirements, which are concerned with the incorporation of user feedback into the underlying context inference processes; and (ii) interaction requirements, which are concerned with displaying context inferences to the user and enabling user feedback.

5.1.1 Feedback Requirements

We begin by specifying a set of abstract requirements for user feedback in interactive intelligent context aware systems. In the previous chapter, we simulated user feedback from the device's perspective using two *actions*: **creation**, where the user can create a new context state from a data observation; and **selection**, where the user can select the correct context state from a list of existing states. Although these actions were suitable for our simulation of user input, a field implementation will need specific and carefully defined feedback requirements from the user's perspective.

In deriving these requirements, we make use of Stumpf *et al.*'s series of experiments [218, 219, 220] that investigate user interaction with machine learning systems. As we mentioned above, there are no standard requirements for incorporating user feedback into machine learning systems, but Stumpf *et al.*'s work is perhaps the most detailed to date. The authors note that telling an intelligent system whether it is right or wrong is a fundamental function for interactive intelligence. They further argue that richer interaction should be encouraged, and we therefore extend the basic functionality.

Table 5.1 summarises our high level user feedback requirements and relates them to user needs; they are further developed in the following list of verb *actions*:

- **Create:** As we discussed in Chapter 3, context data can be sensed from a range of sources, e.g. people, devices and environments. However, following our FSM model of context dynamics in Chapter 4[1], the users themselves should also be able to tell the system about new context states (which the system has not yet seen). Indeed, user-entered data could contain richer context *meaning*, at the cost of elicitation, than many automated context inference processes could potentially output. The key user needs here are **knowledge transfer** – communicating human knowledge to the system – **control** and **personalisation**. The create action is one of the most widely used in Stumpf *et al.*'s interactive email classification studies [220] (though it is termed 'Add').

  Of course allowing users to create their own names for states may have its own difficulties. For example, the user may not be able to think of a good name and may use a more generic, less meaningful one for want of brevity. Thus, for user feedback in mobile context aware systems, we should allow the user to enter data about their context state at any given time. This is the requirement for creation: a mobile context aware system must allow the user to create (using a human-readable label) meaningful context states at any given time. That is, when we design for user feedback, we must design user interfaces to allow context creation.

- **Confirm:** The act of confirming the output of a context inference process forms part of the selection action that we used during active learning in Chapter 4. We distinguish between *confirmation* and *correction* functions, as they have different implications: confirmation implies that the inferred context state is correct and requires only external verification from the 'oracle', i.e. the user, whereas correction implies that the inferred context is incorrect and requires the additional input containing correct context data. The user needs that this requirement addresses are, again, **knowledge transfer** and the **long term reduction in user burden**. As Acid *et al.* [3] and Stumpf *et al.* [220] show, users are willing to provide feedback if they think it will help improve system performance and reduce burden in the long term. Thus, when necessary, a mobile context aware system

[1] Section 4.2.2, page 105

ID	Action	Description	User Needs
II.1	Create	The system must allow the user to create and label meaningful context states.	Knowledge transfer, control and personalisation [220]
II.2	Confirm	The system must allow the user to provide confirmation of correctly inferred context states.	Knowledge transfer and long term reduction in user burden [3, 220]
II.3	Correct	The system must allow the user to correct incorrectly inferred context states.	Knowledge transfer, long term reduction in user burden and improvement of user experience [3, 139, 220]
II.4	Delete	The system must allow users to delete context states.	Privacy and control [140]
II.5	Reset	The system must allow users to reset learned context states.	Long term reduction in user burden and improvement of user experience [3, 139, 220]
II.6	Relabel	The system must allow users to change context state labels.	Privacy and control [140]
II.I	Input	The system must allow the user to perform the feedback functions using one or more input modes	Interaction
II.O	Output	The system must communicate inferred context to the user using one or more forms of output media	Interaction
II.C	Certainty	The system must communicate a measure of confidence in its inferred context to the user.	Intelligibility and control [9, 10, 139]
II.S	Simplicity	The system must satisfy the feedback requirements efficiently, i.e. with as few interactions as possible.	Usability and trust [113]

Table 5.1: *Table summarising the abstract requirements for interactive intelligence in a mobile context aware system.*

must allow the user to provide confirmation of correctly inferred context states.

- **Correct:** The requirement for correction extends that of confirmation. Not only should a mobile context aware system allow the user to provide confirmation of correctly inferred context states, it should also allow them to indicate when inferences are incorrect. Furthermore, it should allow users to supply context data that they (the user) consider to be correct. It is important to note that correction implies that the system has observed the correct context state in the past, or should – for whatever reason – be able to infer a previously unseen state. Previously unseen states that are non-trivial to infer require *creation*. The key user needs here are **knowledge transfer**, as the user is communicating their knowledge to the system through the act of correction, **long term reduction in user burden**, as the system learns from user feedback and reduces the need for user intervention [219] and **improvement of the user experience**, where the user benefits from a more intelligible system [140].

- **Delete:** There may be occasion in which users do not want their context data to be stored or used in future inference processes. This may be, for example, due to privacy or functionality issues, and we should – wherever possible – allow users to have control over their data. The delete action is also one of the most widely used by Stumpf *et al.*'s participants [220]. There is also a need for user **privacy and control**, particularly in relation to intelligibility, as Lim and Dey point out [140]. Of course, the user deleting a state may have an effect on the system: if there are dependencies that rely on that state, they must be addressed in post-deletion updates. Moreover, the question of whether historical records should be erased is important, particularly for privacy reasons, which may have caused to user to delete the state in the first place. Thus, a mobile context aware system must allow users to delete context data, particularly in accordance with any data protection legislation.

- **Reset:** When we integrate learning into the context inference processes as we did in Chapter 4, there comes the risk of error propagation, i.e. an unchecked incorrect inference is used in the learning process which, in turn, affects future inference processes. Although it is prudent to design inference processes to avoid this, we must accept the risk that errors are likely to happen, and we should therefore design a function that allows users to 'reset' corrupted context states. This differs from deletion as the key context data, e.g. the original context label, is not removed entirely. As with the correct action, the user needs are **long term reduction in user burden** and general **improvement of the user experience**. The need for 'unlearning' is directly raised by Stumpf *et al.* in their interactive email classification studies [220].

- **Relabel:** The meanings, or abstract 'labels' of users' context states may change over time, e.g. a location that is used for meetings may sometimes be used for socialising. We must therefore cater for this variability and allow users to relabel context states at any given time. As with deletion, the user needs here are **privacy and control** [140].

5.1.2 Interaction Requirements

Here we define a set of interaction requirements which allow us to enable user feedback in mobile context aware systems. Beyond the primary input/output interaction functions, we consider the system's communication of certainty to the user – which makes particular use of Lim and Dey's work [139, 140, 141, 142, 143] on intelligibility in context aware systems – and simplicity, i.e. minimising the user burden associated with interaction.

- **Input**: To enable user feedback, we must design a set of interfaces that allow users to perform the feedback functions using one or more input modes, e.g. using a keyboard, touch screen or audio input. We should carefully consider which input modes support which feedback functions, whilst keeping the requirement for simplicity (see below) in mind. The user need here is for general interaction, specifically the ability to perform the aforementioned feedback actions.

- **Output**: To allow the system to communicate its inferred context states to users, we must allow communication over one or more output media channels, e.g. audio, visual or tactile feedback channels. We should consider how and when requests for feedback should be made to users. Again, the user need is for general interaction, specifically the ability to perceive the system's demand for feedback actions.

- **Certainty**: Intelligibility is the process of providing the user with explanations as to what an automated system is doing [143, 159]. Intelligibility requirements are important as failure to satisfy them can lead to trust issues between the user and the automated system. Lim *et al.* [141] have adapted a set of intelligibility facets in interface design from Dourish *et al.* [63] and presented a recommended set of design considerations. These considerations form the basis of our user needs: **intelligibility** and **control**. From these, we consider the communication of *certainty* in the system's inference process, i.e. informing the user of how confident the system is in its inferred context. Displaying inference certainty to the user is a tacit method for eliciting feedback in context aware systems [9, 10, 139].

- **Simplicity**: The requirement for simplicity in user interface design has been championed as a positive attribute [113], particularly in relation to **usability** and **trust** in user-centric systems [225]. In our case, we specify the simplicity requirement to 'counterbalance' the idea of intelligibility, i.e. we do not wish to overload the user with unnecessary detail on how the system is working or why it is doing what it is doing. The onus should be on fulfilling the feedback requirements with minimal burden to the user, and simplicity plays an important role in this aim.

In the next section, we return to our concrete use case of place awareness, where we use the requirements from this section to design and implement the components of our interaction layer, namely: the user interfaces for enabling feedback and displaying inferred context; and the output media used to request feedback from users.

5.2 Study: An Interactive Intelligent Place Awareness System

In this section, we outline the development of our mobile place awareness system for deployment in the field. We first state the low level research questions for study and review work that is directly relevant to ours. We then describe the concrete design of the mobile place awareness system, as well as the practical issues involved in development for the Android mobile operating system.

We then describe the approach to our field study, which consists of a 2 week deployment with 10 participants, followed by a *post hoc* place inference performance test. We present the key results and findings before analysing how we have addressed our research questions in light the findings. Finally, we summarise a list of recommendations for designing mobile context aware systems with interactive intelligence, before providing suggestions future work.

5.2.1 Research Questions

Here we develop the low-level research questions for the field study. The findings obtained from analysing a field deployment will help to answer these questions and the mid-level questions of the chapter.

RQ 4.1: To what extent can we infer and actively learn about users' meaningful places using mobile devices?

For a place awareness system to be useful, it should capture as many of the user's meaningful places as possible. Moreover, it should learn about these places each time the user revisits them. There are two key problems surrounding this goal, however: (i) how do we capture meaningful places in the first place? Some places may be important to the user, whereas others – although arguably places – may be unimportant; and (ii) how do we learn about places on revisit? If we wish to use a place awareness system in real time, the system should learn using new location observations as close to when they are observed as possible.

These problems are mainly tackled by the inference and learning work in the previous chapter, and this is the same research question as addressed there. How many meaningful places are captured by the typical user? What characterises the places? Can we visualise what the system is doing when it learns about places? To answer the research question, we need to both analyse and visualise users' place data.

The key question surrounding place awareness is recognition performance: how well does the system infer places when observing new location data? To measure such performance, we need a set of test data and labels for the test data. As we saw in previous chapters, however, observations are usually difficult to capture reliably in field studies (cf. [22, 51, 52]). Thus, to get an indication of classification performance in the field, we need to design a test procedure that reliably captures both observation data and representative test data, i.e. test data that would typically be observed in the field. By using standard classifier performance metrics, e.g. precision, recall, F1 score and accuracy, we can analyse classification performance systematically.

RQ 5.1: How do different user feedback requests affect feedback response rate and time in a mobile environment?

Given the requirements for user feedback, how do we interact with the user in order to elicit feedback at the right time? The key problem here is that users – particularly in a mobile environment – are unlikely to intervene in a background inference processes voluntarily without being requested to, or without incentive. Moreover, even if they are willing to intervene, they are unlikely to do so at the ideal time, i.e. the moment the inference is made. In the mobile environment, the probability of users intervening will almost certainly vary according to their context, e.g. users are less likely to check their device in a bustling shop than at their (typically quieter) desk. Indeed, user behaviour also varies such that we cannot reliably predict if or when users will respond to a request for feedback in a given context.

Given these problems, can we observe how user feedback behaviour, i.e. users' responses to feedback requests from the system, varies according to different request approaches, e.g. using different output media and timing strategies?

RQ 5.2: How do different audio prompts affect feedback response rate and time in a mobile environment?

Interrupting users with audio prompts for input is a popular strategy for data elicitation in ubiquitous computing, but interruptions are annoying and lack of interruptions may lead to missed events [51, 73]. Given the different feedback request approaches in RQ 5.1, can we also study how users respond to different audio prompts attached to certain feedback requests?

As others have noted [75, 237], different approaches to audio prompting can have different effects on user behaviour. How do richer audio prompts such as speech prompts affect feedback response behaviour when compared with more basic audio prompts?

RQ 5.3: How resource intensive is the interactive intelligent mobile context aware system?

One of the key practical problems with mobile context aware systems is the use of mobile resources; particularly power. Employing sophisticated inference techniques and intense sensing policies can potentially use power to the point of making a device unusable [118]. We must therefore consider the usage of power carefully, and constrain both sensing and inference processes accordingly. Balancing energy use against performance is an important trade-off in the development of mobile applications [186], and we should attempt to maximise performance whilst minimising energy use.

Measuring energy use accurately in mobile systems is difficult without specialist equipment [118], and even more difficult to measure in the field. We can, however, estimate sensor usage and infer energy use from sensor use durations [117]. Memory use is also important – and easier to measure – though it is perhaps not as much of an issue as power.

RQ 6.1: What is the user experience like?

Finally, we must consider the question of user experience. Given that interactive intelligence requires engagement of the user, we should attempt to make the feedback process as free from burden as possible without compromising on the integrity of the intended feedback, i.e. it should be difficult for the user to mistakenly enter incorrect data during feedback. To do this requires careful interface design. We can quantitatively measure user feedback through the proportion of actions executed in Table 5.2, but qualitative feedback from users about the experience should also be considered.

RQ 6.2: How does interactive intelligence simulation compare with field behaviour?

One of the questions that arose from our simulation of interactive intelligence in Chapter 4 was the validity of its approach, i.e. how well does the simulation model real user behaviour in the field? It would be interesting to compare the outcome of a field study against the results obtained from simulation to see how valid the simulation approach was.

5.2.2 Related Work

As we saw in Chapter 2, interactive intelligent systems (IIS) are relatively recent subjects of research in AI and HCI [105], with applications such as: text and email classification [219]; database classification [3]; and human-robot interaction [197].

In the mobile domain, recent IISs have been developed in the attempt to predict user interruptibility. Rosenthal *et al.* [196] use a decision-theoretic approach that interrogates users about their interruptibility over time, which draws on similar desktop work by Kapoor and Horvitz [112]. One interesting finding from this study is that overall inference performance can be harmed by asking too *few* questions, i.e. even though there is an interruption cost to the interruptibility questions, this cost does not outweigh the decrease in classification performance gained by prompting the user for input. Another finding showed that overconfident classifiers that are incorrect result in a bad user experience due to the users' perceived intelligibility, i.e. users react negatively to IISs that appear confident yet wrong.

Other work in mobile IISs include Fisher and Simmons [70], who use uncertainty sampling techniques to learn a model of interruptibility for mobile device users. Here the authors show that only a small number of user-supplied labels are required to improve interruptibility classification accuracy in mobile devices. To the best of our knowledge, nobody has yet developed an IIS for mobile place awareness. Our work therefore extends the IIS research field by contributing this.

The key literature for mobile place awareness has been covered in Section 4.3.1 in

Chapter 4 [2], but we detail how the work in this chapter directly compares with the place awareness literature. One of the key points to note is that – unlike the state of the art fingerprint-based methods where both researchers' and infrastructure conditions are enforced, e.g. room-level places within a few seconds' walk are considered a single place [117], labels are predetermined [155] and places are fixed by sets of WiFi access points [116, 115] – we adopt a purely organic approach that allows the users to define their places without restriction (similar to the approach used in the OIL system [173]). We feel this is a fairer representation of users' places, and inference accuracy should measure how well the inference matches the *users'* interpretation of the test data.

Another point of note is that our approach also uses fewer resources than fingerprint-based methods. The state of the art systems that report their resource usage still require a non-negligible amount of WiFi, GPS and accelerometer use [115, 117] in order to function.

5.2.3 System Requirements

ID	Action	Description
II.1	Create	The system must allow the user to create and label meaningful places.
II.2	Confirm	The system must allow the user to provide confirmation of correctly inferred places.
II.3	Correct	The system must allow the user to correct incorrectly inferred places.
II.4	Delete	The system must allow users to delete places.
II.5	Reset	The system must allow users to reset learned places.
II.6	Relabel	The system must allow users to change the labels of their meaningful places.
II.I	Input	The system must allow the user to perform the feedback functions using one or more input modes
II.O	Output	The system must communicate inferred places to the user using one or more forms of output media
II.C	Certainty	The system must communicate a measure of confidence in its inferred places to the user.
II.S	Simplicity	The system must satisfy the feedback requirements efficiently, i.e. with as few interactions as possible.

Table 5.2: *Table summarising the interactive intelligence requirements for the place awareness system.*

In this section, we specify the user feedback and interaction requirements for the mobile place awareness system. These requirements will be used to design the user interfaces

[2] Page 112

within the next section, and they stem from the abstract requirements outlined in Section 5.1.[3]

Feedback Requirements

In Section 5.1, we developed a set of high level requirements for interactive intelligence in mobile context aware systems. In this section, apply these requirements to mobile place awareness. Table 5.2 contains the set of concrete feedback requirements which will form the basis for our system design and subsequent field study.

For place creation, we require a label from the user and a location observation from a set of location sensors as we did in the simulations of Chapter 4. The 'create' action must therefore *allow the creation and labelling of meaningful places.* This defines the requirement for the 'create' action.

Confirmation is the act of telling the system that it is correct, i.e. the place inference that it is not confident about is correct. Correction is the act of telling the system that it is incorrect, i.e. the place inference – regardless of the system's confidence – is incorrect, and supplying it with the correct place. The 'confirm' action must therefore *allow the user to confirm low-confidence place inferences* and the 'correct' action must *allow the user to correct an incorrectly inferred places.*

The 'delete' action is important as users may wish to delete places for a number of reasons, e.g. they don't visit certain places very often, or they don't want their devices to be aware of particular places. Thus, the 'delete' action must *allow users to remove places* from their captured set.

The 'reset' action is important in the context of place awareness, as there is a risk of users mistakenly executing incorrect feedback actions, e.g. confirming an incorrect place when intending to correct it, or anomalous location observations corrupt the learned place models. In these cases, we should implement the reset function in such a way that users can effectively regenerate places, i.e. the 'reset' action must *allow users to reset learned places.*

Finally, the labels of places are extremely important; they serve as human-readable indicators of place inferences and – as they are user defined – they should be editable by the users themselves. Thus, in the context of place awareness, the system must *allow users to change the labels of their meaningful places.*

Each of these actions, except for relabelling – affect the automated inference process in the place inference and learning service, and they form the functional interface between

[3] Page 150

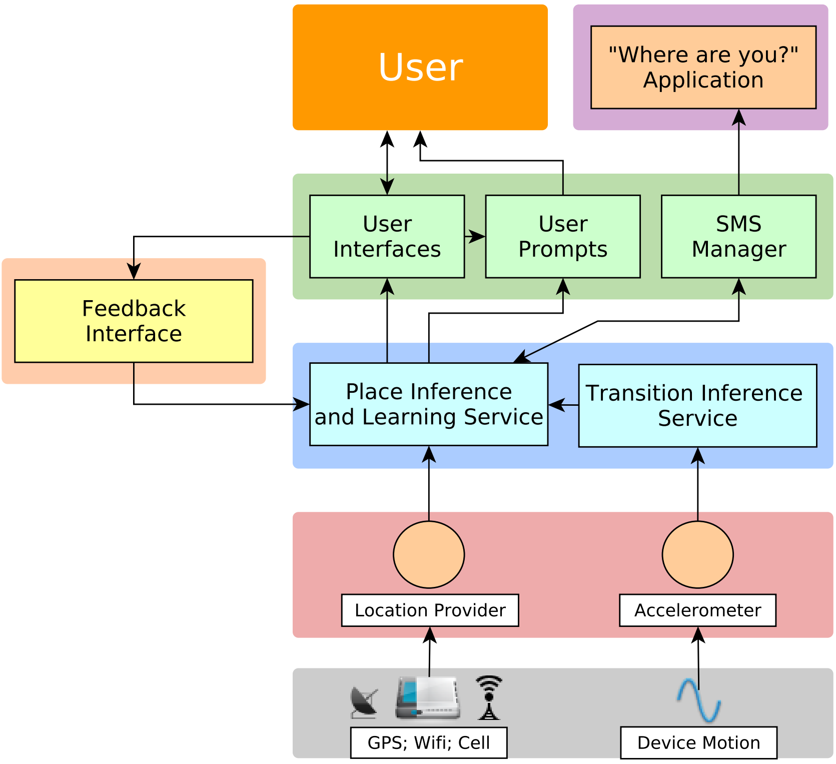


Figure 5.2: *Our layer model illustrating the high level architecture of the place awareness system.*

the user and the place awareness system.

Interaction Requirements

The remaining requirements listed in Table 5.2 are concerned with enabling the user feedback actions in our mobile place awareness system through user interaction. They are effectively the same as their abstract counterparts in Table 5.1.

Table 5.2 contains the full set of interaction requirements that will inform our design of the mobile place awareness system. The requirements are coded with a unique ID so that we can refer back to them as rationale in the next section.

5.2.4 System Design

The high-level layer architecture for the place awareness system is shown in Figure 5.2. This section will describe the design and implementation of each component in the architecture. For the field deployment, the system is implemented using the Android mobile device operating system, versions 2.3 and above.

Context Sensing

As in the previous chapter, the data sources we use are: device motion, sensed through the on-device accelerometer; and location latitude-longitude coordinates provided by Android's location provider service (which fuses GPS, WiFi and cellular data sources). Following energy saving recommendations in [117], we duty cycle the accelerometer to 50% using a period of 30s.

In addition to the sensing capabilities from the previous chapter, we introduce a new GPS speed rule for motion detection. As we saw in the previous chapter, the accelerometer motion models that are trained on walking patterns do not work well when users use other modes of transport, e.g. a car, bus or train. However, by using GPS speed, we can address this issue as GPS typically operates well in most forms of outdoor transport. We therefore sense the user as 'moving' if – when the transition inference system is not detecting the user as 'moving' – the GPS speed is reported to be $> 1.5ms^{-1}$, which is the upper confidence interval (CI) of the empirically observed mean walking speed for humans [54].

Intelligence: Context Inference and Active Learning

At the intelligence layer, we implement both the transition inference and place recognition inference algorithms (Algorithms 6 and 7) from the previous chapter as on-device services: the *transition inference service* and *place inference and learning service* respectively. These are programmed in Java for Android and are designed to be 'always-on' when possible. As Figure 5.2 shows, the transition inference service depends upon the accelerometer sensor and the place inference and learning service depends upon the location provider. The transition inference service outputs to the place inference and learning service alone, i.e. transitions trigger place inferences.

For place inference we use the Bayesian Classifier (BC) from Chapter 4, and we infer new, previously unvisited places using the Mahalanobis distance:

$$D(\bar{\mathbf{x}}, \bar{\mathbf{c}}_p) = \sqrt{(\bar{\mathbf{x}} - \bar{\mathbf{c}}_p)^{\mathrm{T}} \mathbf{S}_c^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{c}}_p)} \tag{5.1}$$

Where $D(\bar{\mathbf{x}}, \bar{\mathbf{c}}_p)$ is the Mahalanobis distance between the location observation $\bar{\mathbf{x}}$ and the weighted mean of place p, $\bar{\mathbf{c}}_p$. $\mathbf{S_c}$ is the weighted covariance of the place's codebook vectors (weighted by sample count).

To distinguish between known places and previously unseen places, we use the Mahalanobis distance corresponding to the χ^2 distribution (1 df) value of 6.64, which corresponds to the statistically significant p-value of 0.01. Thus, location observations

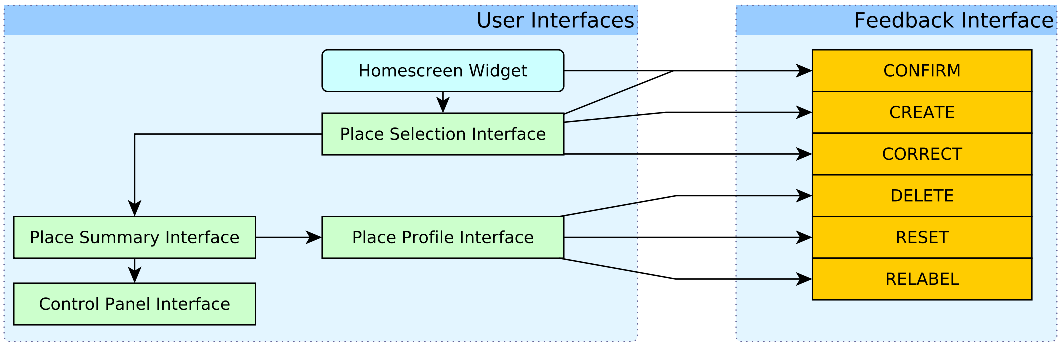


Figure 5.3: *The mapping between user interfaces and the feedback interface. Also shown are the connections between user interfaces, i.e. an incoming edge to interface B from interface A implies that B can be accessed from A.*

made outside this threshold for any place are inferred to be from a potentially new place (which will trigger a 'new place' suggestion – see the next section).

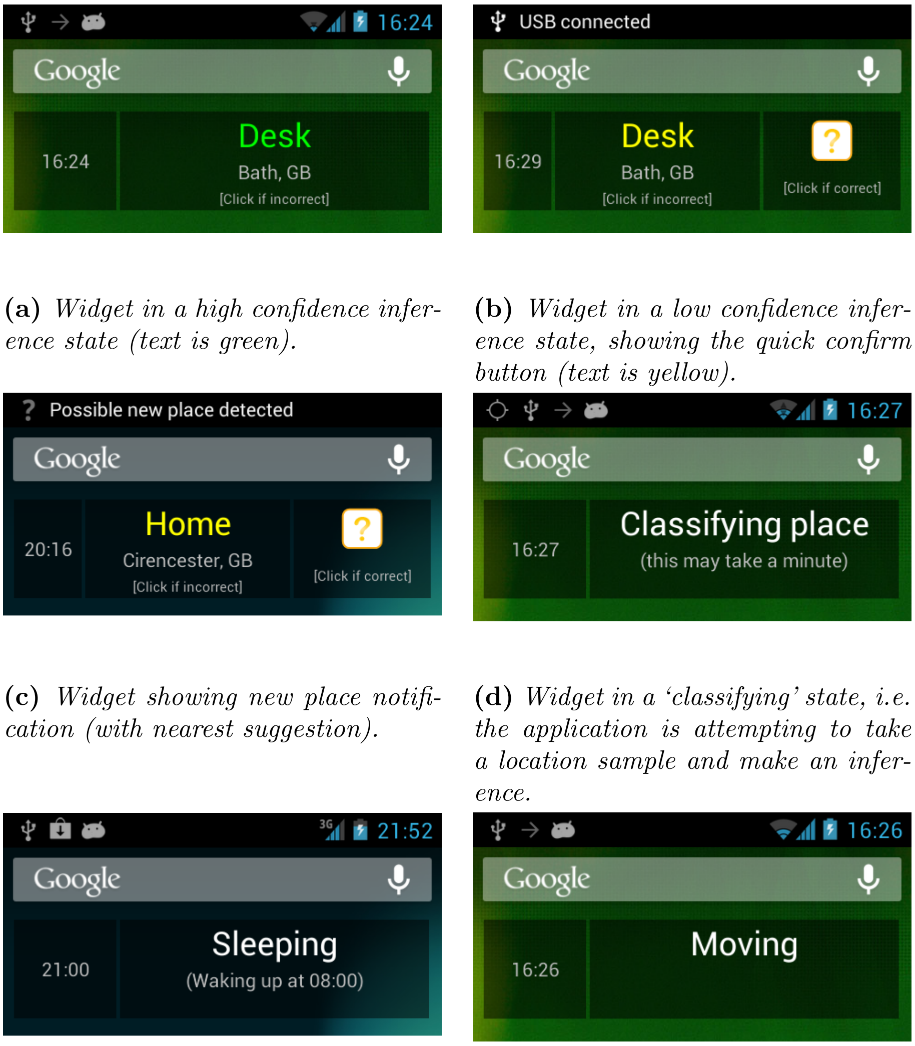
For this implementation, the place inference and learning service also has a periodic update timer which triggers the inference and learning process, i.e. Algorithm 6 from Chapter 4 every t minutes. This is implemented in case place transitions are missed by the detection service and recent location data is required for place inference.

User Interfaces

This chapter has introduced the interaction layer of our layer model, outlining how we might elicit user feedback in the place inference process. One of the most important design considerations in any user-centric system is the user interface, as it can affect many different functionality and experience factors such as the user experience. In this work, the user interface is also the interface that will enable the users to intervene in the place inference process, so it must be carefully designed to fulfil the feedback and interaction requirements in Table 5.2.

We present the primary user interface components, highlighting which of the requirements they are designed to satisfy. In addition to the primary user interface, we will also describe a series of secondary interfaces which will allow the place awareness system to be configured prior to and during the field deployment.

Here we will use the requirements that we specified to design the user interface for the mobile place awareness system. Each interface description will refer back to the requirements in Table 5.2 that it concerns using the requirement IDs from the table. Figure 5.3 shows the mapping between the interfaces and functional requirements from the table, as well as the connections between the interfaces themselves.



(a) *Widget in a high confidence inference state (text is green).*

(b) *Widget in a low confidence inference state, showing the quick confirm button (text is yellow).*

(c) *Widget showing new place notification (with nearest suggestion).*

(d) *Widget in a 'classifying' state, i.e. the application is attempting to take a location sample and make an inference.*

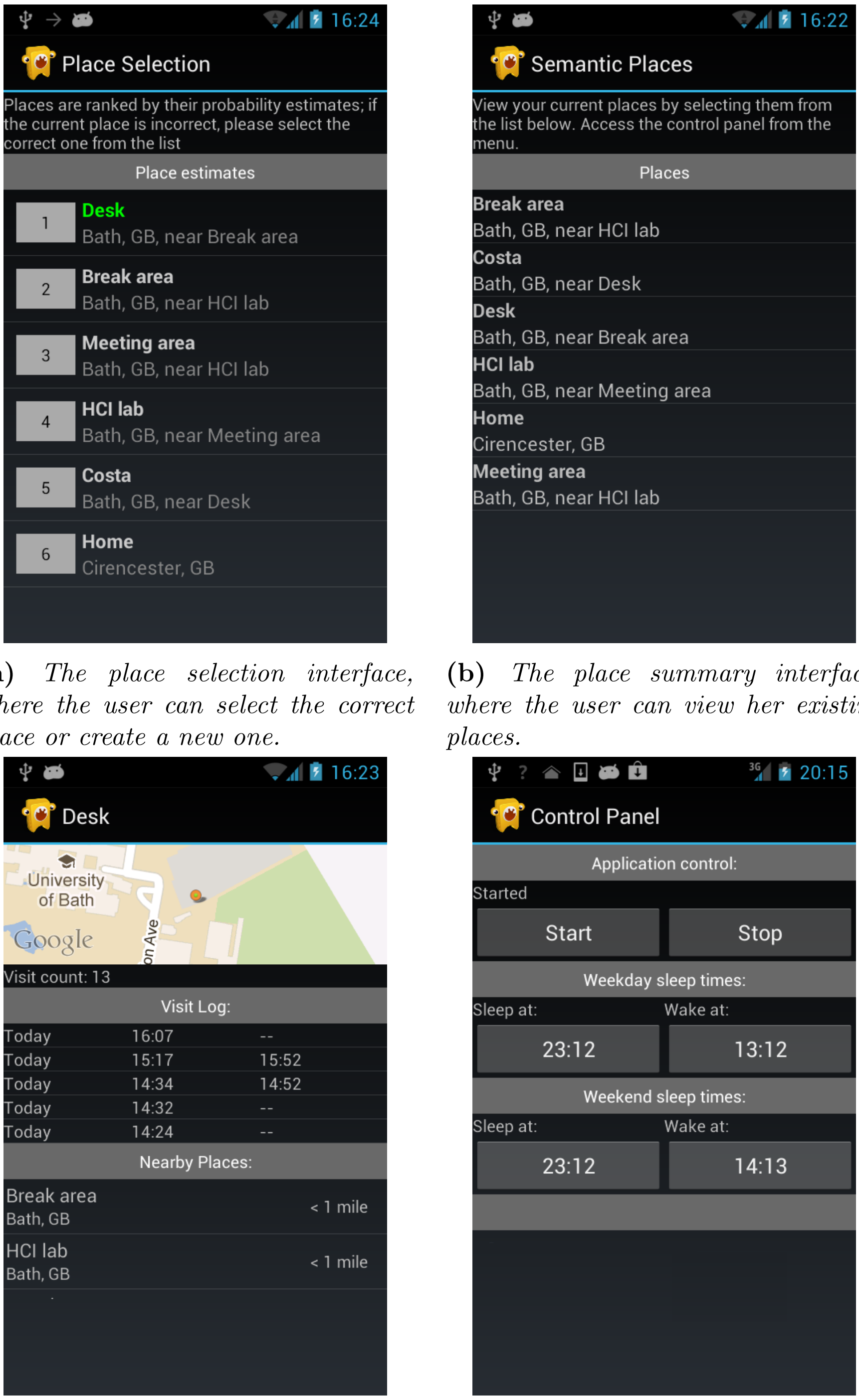
(e) *Widget in a sleeping state.*

(f) *Widget in a moving state.*

Figure 5.4: *Various home screen widget states.*

1. **Home Screen Widget (II.2, II.I, II.O, II.Cand II.S)**: Many mobile devices allow lightweight 'widgets' to operate on their home screens. The majority of Android devices have this feature, and we use it as a miniature user interface to enable quick place feedback. Figure 5.4 shows the home screen widget design for our system; the widget can be in one of seven states, and colour codes communicate inference *certainty* to the user:

   - *High confidence place inference* (Figure 5.4a): The place inference and learning service is confident in its current inference; this confidence is indicated by green text. The user can intervene by clicking on the widget – this brings up the place selection interface in Figure 5.5a.



16:24

Place Selection

Places are ranked by their probability estimates; if the current place is incorrect, please select the correct one from the list

Place estimates

1	**Desk** Bath, GB, near Break area
2	**Break area** Bath, GB, near HCI lab
3	**Meeting area** Bath, GB, near HCI lab
4	**HCI lab** Bath, GB, near Meeting area
5	**Costa** Bath, GB, near Desk
6	**Home** Cirencester, GB

(a) *The place selection interface, where the user can select the correct place or create a new one.*

16:22

Semantic Places

View your current places by selecting them from the list below. Access the control panel from the menu.

Places

Break area
Bath, GB, near HCI lab

Costa
Bath, GB, near Desk

Desk
Bath, GB, near Break area

HCI lab
Bath, GB, near Meeting area

Home
Cirencester, GB

Meeting area
Bath, GB, near HCI lab

(b) *The place summary interface, where the user can view her existing places.*

16:23

Desk

Visit count: 13

Visit Log:

Today	16:07	--
Today	15:17	15:52
Today	14:34	14:52
Today	14:32	--
Today	14:24	--

Nearby Places:

Break area Bath, GB	< 1 mile
HCI lab Bath, GB	< 1 mile

(c) *The place profile interface, giving information about the selected place.*

20:15

Control Panel

Application control:

Started

Start | Stop

Weekday sleep times:

Sleep at: 23:12

Wake at: 13:12

Weekend sleep times:

Sleep at: 23:12

Wake at: 14:13

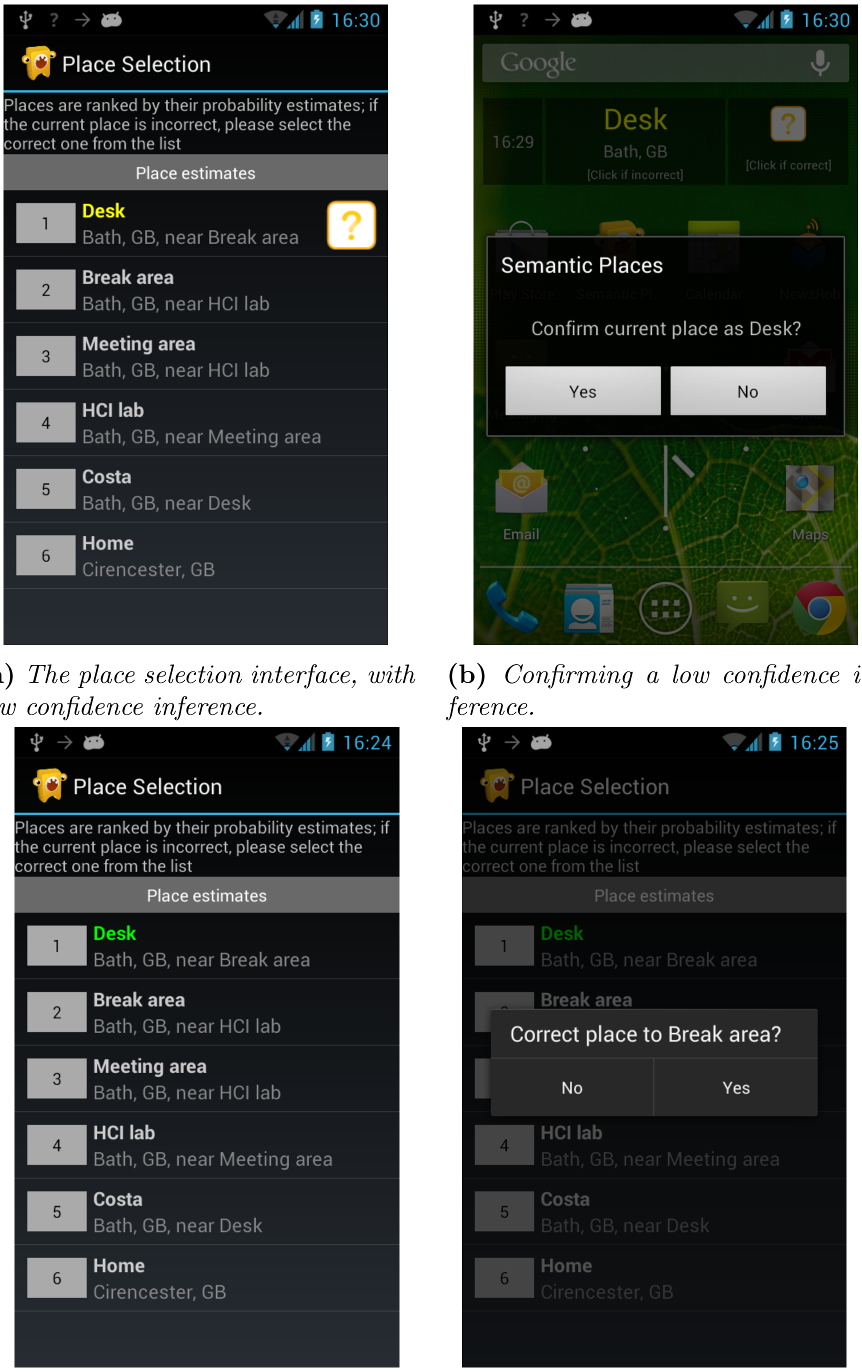
(d) *The control panel, where the user can control system operation; set sleep times; re-train their motion model and edit their approved contacts for the "Where are you?" application.*

Figure 5.5: *Key information interfaces, which enable the user to intervene in place inference and view information about their existing places.*

- *Low confidence inference* (Figure 5.4b): The place inference and learning service is not confident in its inference. The user can click on the question mark icon in Figure 5.4b to quickly confirm the inference as correct (requirement II.2), or they can select the widget text to bring up the place selection interface for correction or creation.
- *New place inference* (Figure 5.4c): The place inference and learning service thinks the current place is previously unseen. A notification is displayed on the device's status bar and the nearest place is displayed in the widget for quick correction.
- *Classifying* (Figure 5.4d): The place inference and learning service is executing Algorithm 6, which is either triggered by the transition inference service or on a periodic update. During this process, the place inference and learning service attempts to sample a set of location data; this can take a variable amount of time or fail entirely, therefore a timeout period, T_L is set.
- *Sleeping* (Figure 5.4e): The system is in sleep-mode, the times of which can be set by the user (see Figure 5.5d). During this state, the system is effectively shut down in order to preserve battery power.
- *In motion* (Figure 5.4f): The place inference and learning service has received a notification that the device is in motion; either from the transition inference service or from GPS satellites, e.g. when driving.
- *No location*: The system cannot obtain a location observation within time T_L and therefore no place is inferred.

All updates to the widget are timestamped and displayed in a text field on the widget.

2. **Place Selection Interface (II.1, II.2, II.3, II.I, II.O, II.Cand II.S)**: (Figures 5.5a, 5.6a, 5.6c, 5.6d, 5.7a and 5.7b) The place selection interface displays the ranked list of places for the current place inference, ranked by probability. If the inference is incorrect, users can select the correct place – if it exists – from the ranked list, or they can create a new place with a label. The user can also confirm low confidence inferences from this interface (as with the home screen widget above). Probabilistic ranking is used in the attempt to minimise scrolling (II.S), i.e. if the correct place is near the top of the ranked list, the user does not have to look far. We use address labels obtained from a geo-location service coupled with the label for each place's nearest Euclidean neighbour as a sub-label in order to defend against ambiguity, i.e. multiple places with the same label.
3. **Place Summary Interface (II.Iand II.S)**: (Figure 5.5b) The place summary interface displays an alphabetically ordered list of the places the user has created



16:30

Place Selection

Places are ranked by their probability estimates; if the current place is incorrect, please select the correct one from the list

Place estimates

1	**Desk** Bath, GB, near Break area
2	**Break area** Bath, GB, near HCI lab
3	**Meeting area** Bath, GB, near HCI lab
4	**HCI lab** Bath, GB, near Meeting area
5	**Costa** Bath, GB, near Desk
6	**Home** Cirencester, GB

(a) *The place selection interface, with low confidence inference.*

16:30

16:29 Desk Bath, GB [Click if incorrect] [Click if correct]

Semantic Places

Confirm current place as Desk?

Yes No

(b) *Confirming a low confidence inference.*

16:24

Place Selection

Places are ranked by their probability estimates; if the current place is incorrect, please select the correct one from the list

Place estimates

1	**Desk** Bath, GB, near Break area
2	**Break area** Bath, GB, near HCI lab
3	**Meeting area** Bath, GB, near HCI lab
4	**HCI lab** Bath, GB, near Meeting area
5	**Costa** Bath, GB, near Desk
6	**Home** Cirencester, GB

(c) *The place selection interface, with high confidence inference.*

16:25

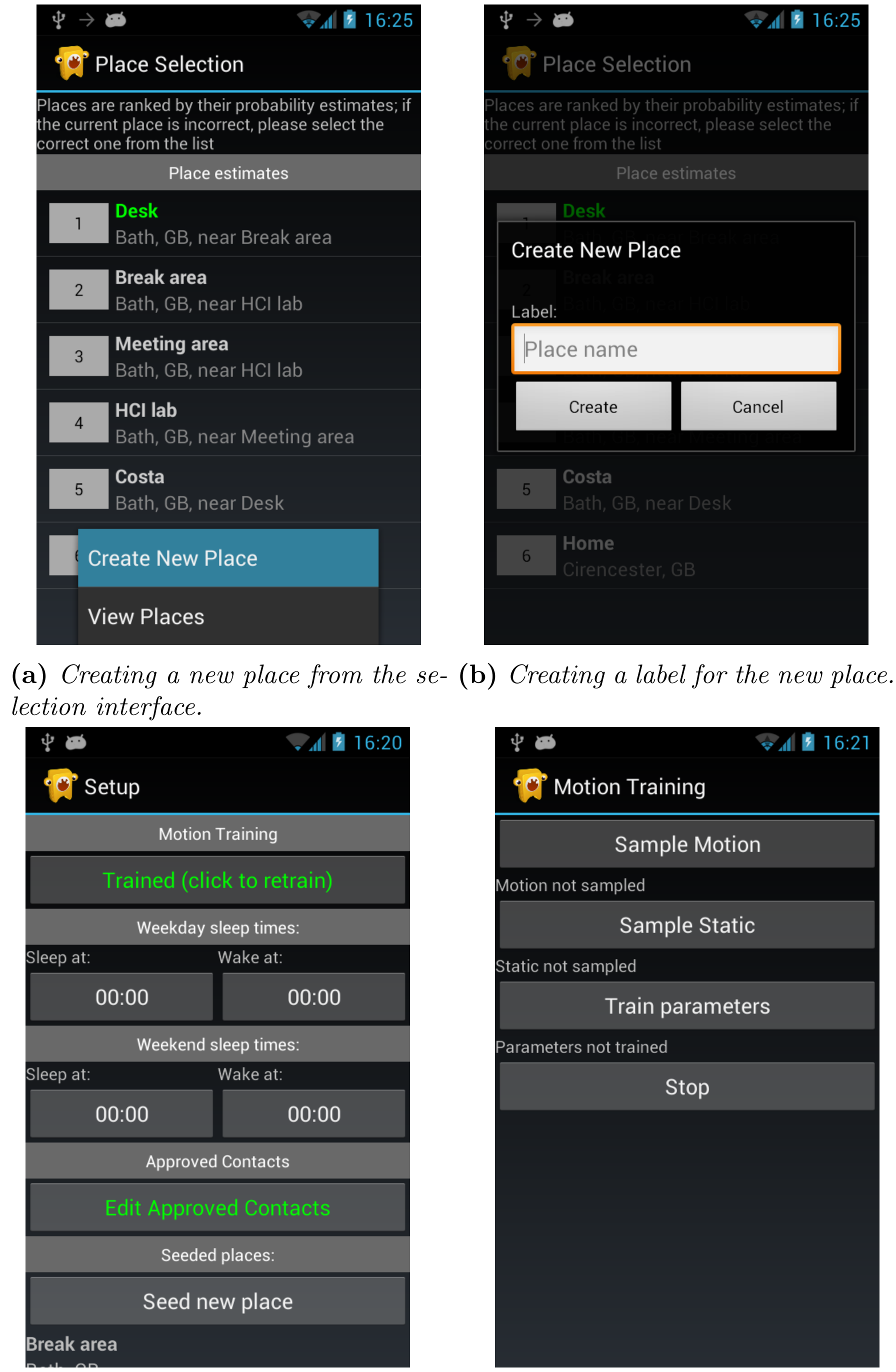
Place Selection

Correct place to Break area?

No Yes

(d) *Correcting one place to another.*

Figure 5.6: *Confirming and correcting places through the place selection interface*



(a) *Creating a new place from the selection interface.*

(b) *Creating a label for the new place.*

(c) *The 'first run' interface for configuring the motion model, sleep alarms and approved contacts.*

(d) *The motion training interface*

Figure 5.7: *Place creation and first run interfaces*

so far. As with the place selection interface, each place is labelled with the user's label and stamped with a reverse geo-location address (if available) and the label of the nearest Euclidean neighbour in coordinate space. Selecting a place from this list brings up the place's profile interface.

4. **Place Profile Interface (II.4, II.5, II.6, II.Iand II.O)**: (Figure 5.5c) The place profile interface shows: a map view of the place's weighted mean vector; a list of recent visits; and a list of nearby places. From here, the user can re-label the place, reset the location data for the place, i.e. regenerate a fresh set of codebook vectors, or delete the place.

In addition to these key interface components, there are a set of interfaces for configuring the system on first-run. These are:

- **Sleep Alarm Interface**: (Figures 5.5d and 5.7c) Sleep alarms allow the user to set periods of time during which the application will sleep, i.e. temporarily shut down. This allows users to save battery for typically long periods of inactivity, e.g. sleeping. The option to have different sleep times for weekdays and weekends is provided.
- **Motion Trainer Interface**: (Figure 5.7d) This allows users to train their motion model based on sampling both motion and static data for a short period of time. Pressing both the 'Sample Motion' and 'Sample Static' button will launch the accelerometer sampling process, while the objective for the user is undergo a state of motion or 'non-motion' accordingly. Pressing 'Train Parameters' will launch the stochastic gradient descent process for training the logistic regression classifier, which forms the key component for the transition inference service.
- **Approved Contacts**: For privacy reasons, the "Where are you?" application (see Section 5.2.4) allows the user to select a set of approved contacts whom she feels comfortable sharing her current place with.

Finally, there is a control panel interface which allows the user to control the system's operation, including: starting and stopping the system; changing the sleep alarms; and editing their approved contacts for the "Where are you?" application. This is shown in Figure 5.5d.

Feedback Requests and Audio Prompts

RQ 5, RQ 5.1 and RQ 5.2 ask about the elicitation of user feedback and how various approaches might affect users' response to requests for feedback. In the simulations of the previous chapter, we modelled feedback by sampling from a Bernoulli distribution parameterised by participants' estimates of how likely they felt they would respond to a device notification in each given place. Although this is probably a more natural model than pure random sampling, it still does not capture the intricacies and external stimuli that users may be subjected to in the field.

Moreover, during the simulations, we made the assumption that the user was a 'perfect oracle', i.e. when they chose to intervene, they would always give the correct response. Of course there is a risk that they might give an incorrect response, e.g. ambiguous labelling, where two or more places have the same label, or mistakenly pressing confirming a place when they intended to correct it. The place selection interface and the home screen widget are both designed to guard against these events (see Figures 5.5a and 5.6b).

The remaining challenge, therefore, is eliciting feedback from the user when required. Timely requests are extremely valuable for eliciting user feedback [115], and have been shown to improve the perceived burden of interruption from mobile devices [196]. Indeed, actively requesting for user input at the point of transition between context states (specifically activities) is also a useful approach to relieving the burden of user interruption [92].

Following on from our work in Chapter 4 on inferring transitions between places, we design a feedback request subsystem that raises the necessary requests for user feedback *at the moment of inferred transition* between places, i.e. during the NotifyUser subroutine in Algorithm 6.[4] Although transition inference performance in Chapter 4 was good, there is still a risk of transitions being missed. The periodic updates described in Section 5.2.4 are designed to guard against these missed events, and we can also attach feedback requests to these updates.

We differentiate between *requests* for feedback and *prompts* that are attached to requests. Feedback requests are raised by the inference process when it requires user feedback. Prompts are notifications made through various output media in order to make users *aware* of feedback requests. For our study, we compare two request approaches (see Table 5.3 for the relationship between the request types and their prompt media):

- **ACTIVE**: ACTIVE requests are raised during the NotifyUser subroutine in Algorithm 6 *if the algorithm is triggered by a place transition* (as opposed to a periodic update, described in Section 5.2.4). This subroutine is only called in the case of a low-confidence inference or new place inference, so confident inferences will not have requests attached to them. ACTIVE requests will raise audio, visual and tactile *prompts* simultaneously on the device. Visual prompts

[4] Page 128

		Prompt Output Media			Time Policy
		AUDIO	VISUAL	TACTILE	
Request	ACTIVE	x	x	x	On transition
Type	PASSIVE		x		Periodic

Table 5.3: *Table illustrating the mapping between feedback request approaches and prompt output media, as well as the requests' time policies.*

update the homescreen widget and place selection interface, whilst tactile prompts trigger the device's vibration hardware. Audio prompts are discussed shortly.

- **PASSIVE**: PASSIVE requests are raised during the NotifyUser subroutine in Algorithm 6 *when the algorithm is triggered by a periodic update.* As with ACTIVE requests, confident inferences will not raise PASSIVE requests. PASSIVE requests do not make any audio or tactile prompts for user feedback; rather they use visual prompts that simply update the home screen widget to a low-confidence state as in Figure 5.4b. The only indications to the user that a visual prompt has been raised is a change in widget state and an updated widget timestamp.

There are points at which users may intervene without any request by the system, and we do not restrict them in doing so. For the study, we measure how much feedback is given without request, and compare user response behaviour against the ACTIVE and PASSIVE requests.

Audio prompting is a research subject in its own right [75], and it would be interesting to compare different approaches to audio prompting for ACTIVE requests. Many devices nowadays can output speech using text-to-speech (TTS) engines as well as basic notification tones. As such, we compare two types of audio prompt for our ACTIVE requests:

- **TTS**: Audio prompts with TTS allow the device to 'say' the inferred place name. The idea is for TTS prompts to convey more information about the feedback request than basic audio prompts without TTS can. We use the user's own labels in the TTS prompt, i.e. the label used to create a meaningful place.
- **NO TTS**: Audio prompts without TTS use a non-speech tone as an audio reminder.

"Where are you?" – A Presence and Availability Application

The SMS manager in Figure 5.2 manages the SMSs for the "Where are you?" application – the algorithm for which is shown in detail in Algorithm 8. By first checking

Algorithm 8 “Where are you?” SMS algorithm executed upon receipt of an SMS. Comments indicate SMS reponses

```
Input: SMS – an SMS message; $p$ – classification confidence threshold;
  $d$ – distance threshold; $r$ – recency threshold
Output: Response – a response string, to be sent as an SMS to the
  originating contact
contact ← GetContact(SMS)
if IsApproved(contact) = False then
    return
end if
text ← GetContent(SMS)
match ← RegexMatch(SMS, [Ww]h?ere +[Aa]?[Rr]e? +([Yy]o)?[Uu][!\?]*)
if match = False then
    return
end if
response ← “No recent place information, sorry”
place ← GetLatestPlace()
time ← GetExitTime(place)
conf ← GetConfidence(place)
if IsMoving() = True then
    if conf $\geq p$ then
        mins ← GetTimeDifference(time, now, MINUTES)
        response ← MakeResponse(place, mins)    ▷ “Left place mins minutes ago”
    else
        place ← GetLatestConfidentPlace()
        time ← GetExitTime(place)
        mins ← GetTimeDifference(time, now, MINUTES)
        if mins $\leq r$ then
            response ← MakeResponse(place, mins)▷ “Left place mins minutes ago”
        end if
    end if
else if IsPlaceInferred() = True then
    if conf $\geq p$ then
        response ← MakeResponse(place)                              ▷ “place”
    else
        neighbours ← GetNeighbours(place, $d$)
        response ← MakeResponse(place, neighbours)    ▷ “Near place, neighbours”
    end if
else
    mins ← GetTimeDifference(time, now, MINUTES)
    if mins $\leq r$ then
        response ← MakeResponse(place, 0)                ▷ “Was at place recently”
    end if
end if
return response
```

whether the SMS sender is on the approved contacts list, the algorithm will – if the sender is approved – parse the message using a regular expression for variants of the phrase "where are you?". If a positive match is found, the SMS manager creates an SMS response based on current place conditions and sends the response to the originating contact.

The "Where are you?" application is an example of a presence and availability application that uses the mobile place awareness system to share the user's current place with a set of approved contacts if requested by any of the contacts. If instructed by the SMS manager, the "Where are you?" application will respond on receipt of a "where are you?" SMS with an estimate of the user's current place.

The response is built according to a set of conditions:

- If the current place inference is confident, the "Where are you?" application responds with the user's place label and an address stamp (if available).
- If the current place inference is not confident, the application responds with a list of nearby places or – if the distance to the nearest place is greater than a certain threshold d – the current address from Android's geo-location service.
- If the system is in a state of motion, the application responds with a "left <place> x minutes ago", where <place> is the inferred place within the distance threshold d of the most recent location observation. (If the inferred place is outside this threshold, then the application falls back to the first place inference that was within it.)
- If there is no location data for the user, the application responds with a "No recent place information, sorry" message.

Thus, the goal of the application is to communicate the device's current best estimation of the user's place to the closest members of their social network. If uncertainty is present, or there is a lack of necessary data, the application attempts to provide a reasonable estimate of the user's surrounding or recent places. Only when it has exhausted these options does it respond with a message of failure.

Detailed Design Architecture

Figure 5.8 shows the detailed architecture of the place awareness system, highlighting the inference algorithms, the device interfaces, the feedback interface and the "Where are you?" application.

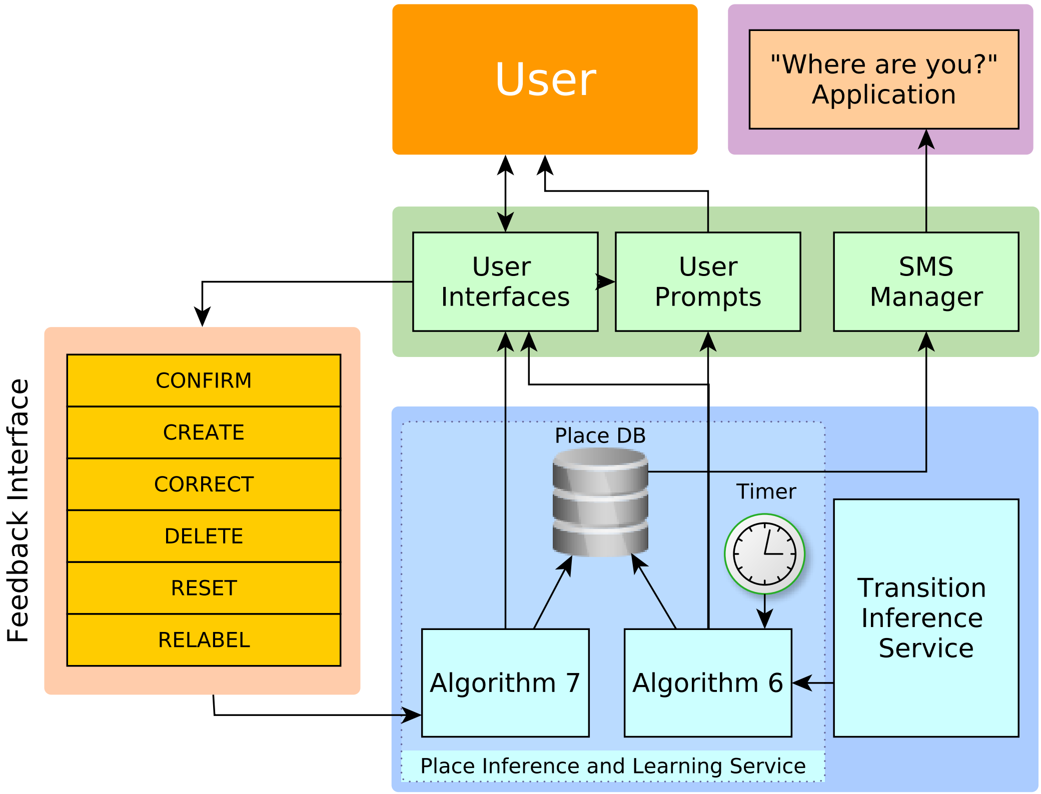


Figure 5.8: *Detailed architecture of the place awareness system's inference, feedback and interaction components. Algorithms 3 and 4 refer to the place inference algorithms from Chapter 4*

5.2.5 Study Approach

Following an initial pilot study of the mobile place awareness system with 5 participants, we recruited 12 new participants for the main study. 2 participants left the study early, leaving a final set of 10 (2 female, 8 male, mean age 29.8; sd 6.6). This is a similar sample size used by previous field work that is closely related to ours, e.g. [17, 156, 155]. The participants were recruited as they owned and were familiar with Android mobile devices; 7 of the participants used their own devices throughout the duration of the study, and 3 (IDs 1, 3 and 6) were provided with test handsets as their Android version was incompatible with the application. These participants used their own SIM cards and Google accounts in the Android test handsets, however. The participants were also split across two key demographics: 4 of the participants were working professionals (IDs 5; 8–10) and 6 were postgraduate computer science students. This was done to vary the participants' daily environments an increase the range of potential places captured, rather than capturing a large number of places from the same environment. A similar split and sample size was used in Barkhuus *et al.*'s Connecto study [17]. To compensate participants for their time and data/SMS usage throughout the study, each were paid

£20.00 upon completion.

The study was undertaken in three phases: a setup and configuration phase, the field deployment and a performance test phase.

Setup and Configuration

For initial setup and configuration, each participant was provided with a consent form detailing the study requirements (Appendix C.1) and a brief instruction manual on how to use the application. The instruction manual contained the Harrison and Dourish example of 'place' vs 'space' [86], and examples of places from Barkhuus *et al.* [17], including 'work', 'home' 'café' and 'gym'. As with the place studies in the previous chapter, these differences between place and location – as well as the examples of places – were explained to the participants verbally. Following this, a researcher guided each participant through an example of each feedback action in Section 5.1 on a separate device running a live version of the place awareness system, before starting the participant's own system on their device for the first time.

As the first interface to be presented on initial start is the 'first run' interface in Figure 5.7c, a researcher guided the participant through the necessary initialisation steps. Each participant trained a motion model for the transition inference service as before; by carrying the device as they typically would, e.g. in a pocket or a bag, and recording 30 seconds of motion data when stationary and walking. As before, the parameters for the logistic function were trained using regularised logistic regression using stochastic gradient descent on the mobile device.

Following this, each participant selected a set of sleep times for weekdays and weekends, before marking a list of approved contacts from their device/SIM contacts list in order to initialise the "Where are you?" SMS awareness service.

Participants were then instructed to select at least one 'seed' place to allow the place inference and learning service to initialise. To do this, each participant chose a nearby meaningful place at the time of setup and moved to it. Once there, they pressed the 'Seed new place' button in Figure 5.7c which prompted them for a label before taking a location sample to initialise the seed place's codebook vectors. More details on the instructions given to participants can be found in Appendix C.

Field Deployment

The field deployment ran for 2 weeks, starting immediately after the setup phase described in the previous section. During the field deployment, the participants were instructed to allow the place awareness service to operate continually on their mobile device, i.e. they should avoid stopping, 'killing' or un-installing it. They were *not* told to keep their devices turned on throughout – rather that they should follow their normal behaviour patterns, e.g. turning their device off overnight. As the application used audio prompts even in silent mode, participants were instructed to stop the application through the control panel if silence was absolutely necessary without device shutdown.

To observe feedback behaviour according to the different feedback request types and audio prompts (RQ 5.1 and RQ 5.2), we implemented two between group conditions and two within group conditions as follows:

- **Feedback Request Approach (within)**: All participants were subject to ACTIVE and PASSIVE feedback requests as described in Section 5.2.4.
- **Audio Prompt Approach (between)**: Participants were initially split into two groups of 6 but – following the 2 drop-outs part way through the study – the actual split was 6 and 4. The first group had TTS audio prompts attached to their ACTIVE feedback requests (participants 1, 2, 5 and 10), while the other group had a simple audio notification – the 'Capella' tone on Android handsets – attached to theirs.

The participants were encouraged to choose any label they wished for each place, and no restrictions were made on characters, length or format of the labels. The participants were *not* told that creating places, manually updating places or intervening would correlate with any reward, i.e. there was no goal to 'tag as many places as possible', nor was there any competition between participants to 'check-in' to particular places more than others. Participants were only told that their device would sometimes make requests for feedback – as demonstrated in the setup phase – and sometimes wouldn't, and that it would always display the home screen widget which would sometimes update itself. Participants were *not* told that ACTIVE requests would likely occur on transition into places, nor that PASSIVE requests were periodic. Each TTS condition group was not told about the between group condition.

The place awareness system operated continually on each participant's device throughout the field deployment. There were three occasions in which participants turned their device off for 1 day (IDs 5, 7 and 8), and there were occasional system crashes. However, a crash-detection service immediately restarted the application in each case and no data were lost due to the crashes.

All transitions, requests, classifications, user feedback actions, location samples and sensor activations were logged. All sleep times were logged except for participants 3 and 6, whose sleep data became corrupted during the study and we did not realise until after the study was complete.

Following the field deployment, each participant was asked to (anonymously) complete an exit survey relating to their experience of the mobile place awareness system during the field deployment. Once all surveys had been completed, we interviewed each participant for further recorded feedback on the user experience.

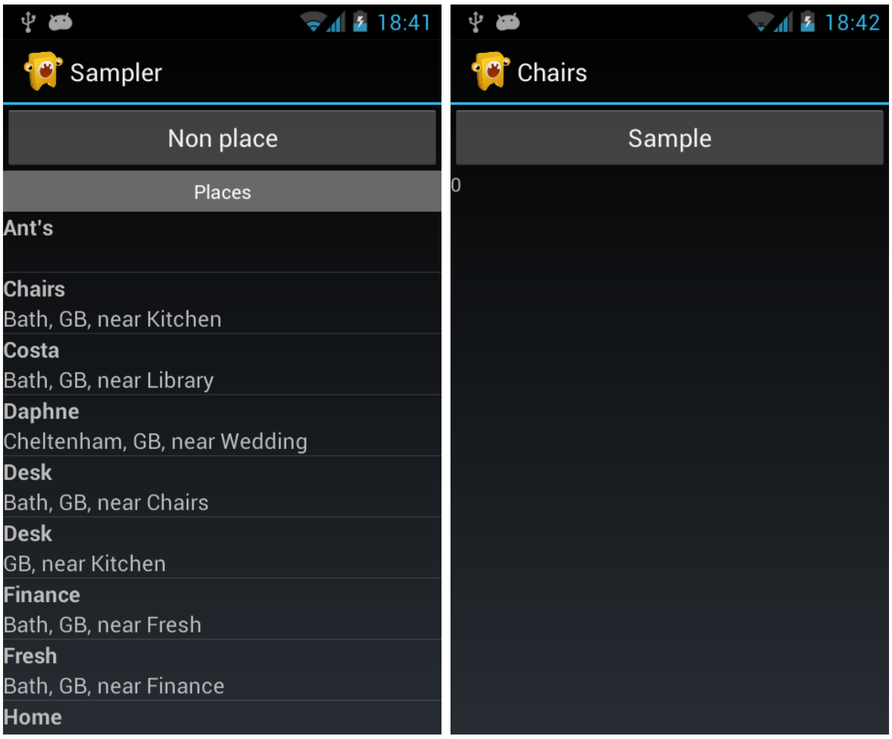
Feedback Behaviour Measures

In order to measure user feedback behaviour and answer RQ 5.2, we use three measures:

- Feedback action distribution: given the feedback functions outlined in Section 5.1, we can measure the distribution of each participant's feedback proportion over these actions. From this distribution, we can infer further properties, e.g. a greater proportion of confirmations than corrections implies that the inference processes are correct and not confident, rather than incorrect.
- Feedback request response rate: by logging when feedback requests are made, and whether they are ACTIVE or PASSIVE, we can also measure how many requests the participants respond to; allowing us to estimate feedback request response rates as a proportion of requests responded to.
- Responded requests' response time: for requests that are responded to, we can measure the time period between when the request was raised and when it was responded to.

Place Inference Performance Measures

In order to test the place recognition performance of the system, we treat it as a statistical classifier and present it with test examples for classification. By comparing these test examples with observations, we can measure inference performance using standard metrics. It is very difficult to capture fine-grained observations in the field [195] and the use of diary studies – in addition to being unreliable – may affect our participants' natural feedback in the field. We therefore do not explicitly analyse any performance metric during field deployment. Rather, we treat the 2-week field deployment as a training phase, after which each participant's places' codebook vectors are modelled as a classifier and presented with previously unseen test data from a *post hoc* scripted tour to classify. Although this approach does not measure performance in fine-grained detail in the field, it does provide an indication of how well the actively learned place models classify new data.



Ant's

Chairs
Bath, GB, near Kitchen

Costa
Bath, GB, near Library

Daphne
Cheltenham, GB, near Wedding

Desk
Bath, GB, near Chairs

Desk
GB, near Kitchen

Finance
Bath, GB, near Fresh

Fresh
Bath, GB, near Finance

Home

(a) *The test application main interface.* **(b)** *The test application sample interface.*

Figure 5.9: *The interfaces for the participants' test application. They allow participants to select a place (or 'non-place') and take a location observation for performance testing.*

For the *post hoc* tests, a small application was developed that presents a list of places to the participant (see Figure 5.9a). The participant can then select a place and take a location reading (Figure 5.9b), which is then stored as a test datum with associated observation. Location data are sampled exactly as in the field deployment, i.e. the device attempts to take 10 samples per observation; the observation is the weighted mean of these samples (weighted by inverse reported accuracy).

Thus, following completion of the field deployment phase – and prior to the *post hoc* test – each participant was asked to identify a set of nearby places without directly referring to the list of places created during the field deployment (in case there were meaningful places nearby that the system did not capture, and to reduce bias in using the list to pick places). They were also asked to identify a set of meaningless areas – or 'non-places' – in between the meaningful places to fully test inference accuracy. The place awareness system was then un-installed from each participant's device, and the database copied offline for analysis. Following this, the test application in Figure 5.9 was installed on the same device, and the participants were asked to go to each of their identified meaningful places and take 5–10 location samples at each using the test application.

To encourage fairness, the participants were explicitly told to take their samples at different points within each place, e.g. different rooms of a building, or different areas of a room. We told the participants that these areas should not be chosen randomly, rather they should be representative of where they would plausibly go within each place. They were also asked to capture as many samples from meaningless places as possible throughout the test. For each sample taken, the participants were instructed to place the device in a natural position, i.e. how they would normally carry it in each area within the current place. A small time delay of 10s from button press to sample was introduced to give the participants time to do this. The full instructions can be seen in section C.1 of Appendix C.

We use the same classification process for testing as we used in the place awareness system, i.e. Algorithm 6 without the confidence measure or codebook vector update sub-routines. Each place is again modelled as a bivariate Gaussian distribution, parameterised by the weighted mean $\bar{\mathbf{c}}$ and weighted covariance $\mathbf{S}$ of the place's codebook vectors (weighted by sample count). For the tests, we use the Mahalanobis distance as a distance measure as in Equation 5.1, and the χ^2 1 d.f. critical value of 6.64 ($p = 0.01$) as a threshold to distinguish between meaningful and unimportant places.

We measure classification performance using four base metrics:

1. True positives (tp): If the classifier outputs a place with the label matching observation given a test location input.

2. False positives (fp): If the classifier outputs a place with a label that doesn't match observation given a test location input.

3. True negatives (tn): If the classifier outputs a 'non-place' matching a meaningless observation given a test location input.

4. False negatives (fn): If the classifier outputs a 'non-place' that doesn't match observation, i.e. the test location datum is labelled with a place label.

These metrics can be encoded into various performance measures, namely:

- Precision:
$$p = \frac{tp}{tp + fp} \tag{5.2}$$

- Recall:
$$r = \frac{tp}{tp + fn} \tag{5.3}$$

- The F1 score, which is the harmonic mean of precision and recall:

$$F1 = 2\frac{pr}{p + r} \tag{5.4}$$

- Accuracy:

$$a = \frac{tp + tn}{tp + tn + fp + fn} \tag{5.5}$$

Size	Description
<ROOM	A place smaller than a room, e.g. an area within a room.
ROOM	A place that is approximately room-sized, i.e., smaller than a building, or described by a physical room.
BUILDING	A place that is approximately building-sized, i.e. the size of multiple rooms, or described by a physical building.
>BUILDING	A place that is larger than a building, i.e. corresponding to multiple buildings or towns/cities.

Table 5.4: *Table summarising the size categorisations of the participants' places.*

Finally, participants were asked to give an estimate as to the approximate size of their places according to the categories in Table 5.4. This was done to provide information on the sizes of places, which are likely to have an effect on classification performance (particularly small places that are in close proximity to each other).

	Label	Tested	CONFIRMED	CORRECTED	SIZE
1	desk	1	66	1	<ROOM
2	breakout	1	0	3	<ROOM
3	printers	1	1	7	<ROOM
4	RnD room 2	1	0	1	ROOM
5	kitchen	0	0	2	<ROOM
6	bike shed	1	0	0	ROOM
7	home	0	15	0	BUILDING
8	coffee shop	1	2	0	ROOM
9	Baird house mtg room	1	0	0	BUILDING
10	Nby train station	0	1	0	BUILDING
11	vf Paddington	0	8	0	BUILDING
12	Restaurant	1	3	0	ROOM
13	nby Waitrose	0	0	0	BUILDING
14	one stop	0	1	0	BUILDING
15	sun in wood. pub	0	7	0	BUILDING
16	Babbage house	1	0	0	BUILDING
17	ground floor breakout	1	0	0	<ROOM

Table 5.5: *Place data captured by Participant* 5 *during field deployment, showing: place labels, feedback actions and size categories*

5.2.6 Results

This section details the results obtained from the field study. We first present results from the places captured during the field deployment, including contour plots of example places, labels, sizes and counts. We then present user feedback results obtained during field deployment, including: automation-supervision results; feedback action results and distributions; and prompt response times. Following this, we present performance results from the post-study performance testing, including sizes and sample counts of the places tested. Finally, we present a set of results on resource usage, location accuracy and results from the post-study exit survey.

The key parameters for the study were instantiated as follows: $T_L = 60$s (the location sampling timeout period); $t = 15$mins (the time period for periodic place updates); $d = 200m$ (the "Where are you?" SMS distance threshold for neighbour places); $r = 10$mins (the "Where are you?" recency parameter). These were chosen heuristically and an interesting avenue for further work could be to explore how varying these parameters affects performance.

Place Capture and Learning

Here we outline the results of the places captured during the study. Table 5.6 shows the number of places captured by each participant during the two-week field study. The median count was 17 (sd = 4.12).

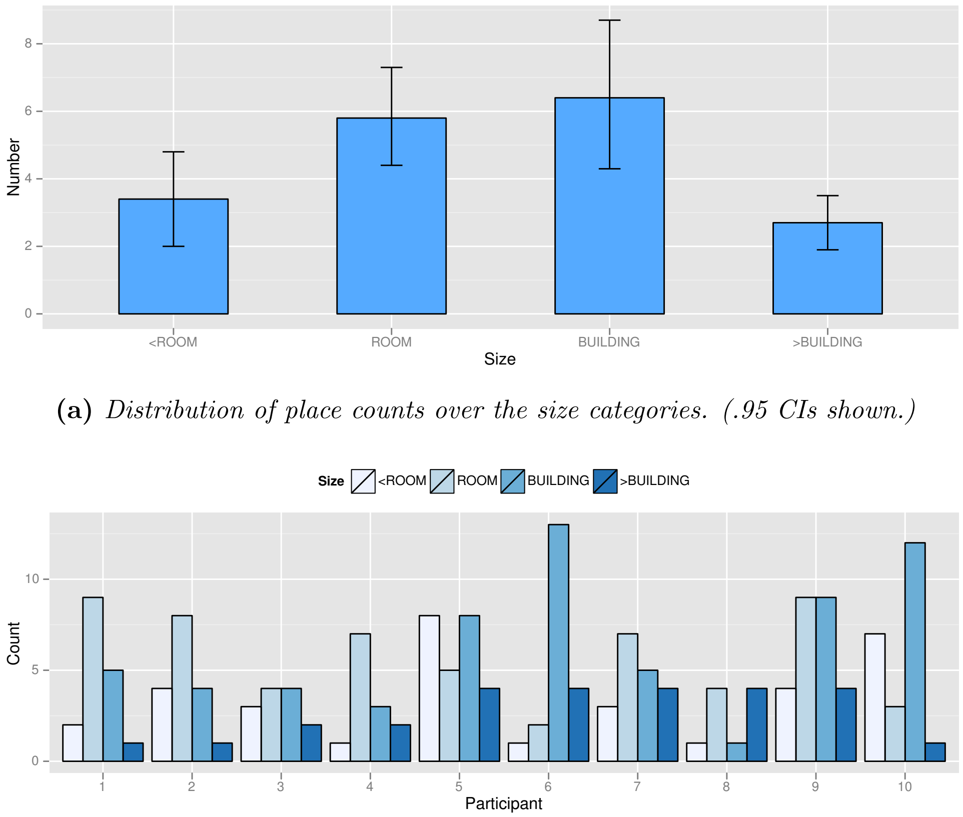
Participant	1	2	3	4	5	6	7	8	9	10
Count	17	17	13	13	17	20	19	9	13	23

Table 5.6: *The place counts over the participants over the 2-week field study*

A full collection of the participants' captured places – including labels, feedback data and size categories – can be found in Appendix C.2, but Table 5.5 shows these data for Participant 5.

Figure 5.10 shows the distribution of place counts over the size categories and participants. The only significant differences are between the BUILDING and >BUILDING categories, and the ROOM and >BUILDING categories ($p < 0.05$; non-parametric bootstrap, 1000 replicates, $N = 10$).

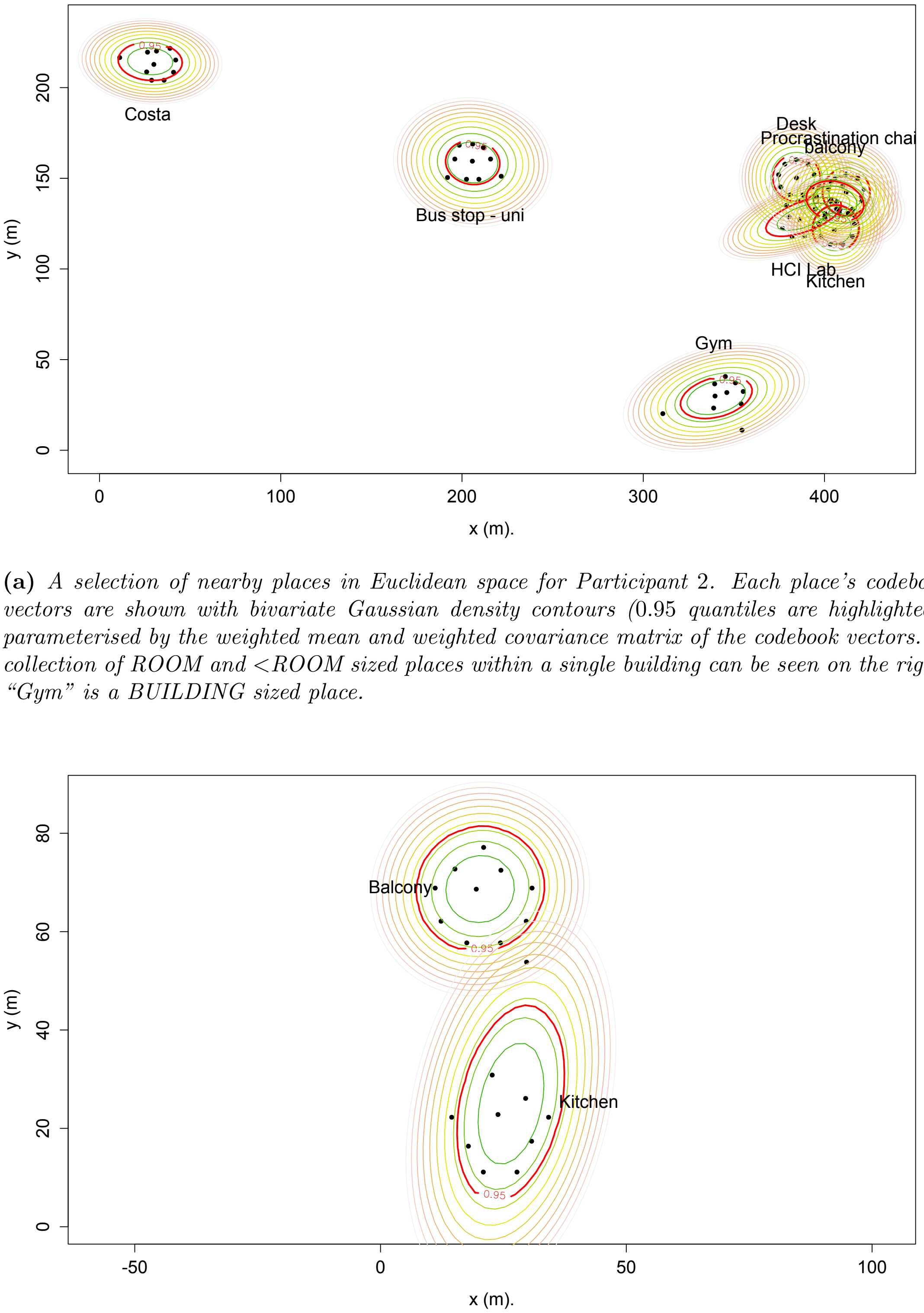
Figures 5.11, 5.12 and 5.13 show contour visualisations of various places for a selection of the participants. The bivariate Gaussian contours are fitted to the codebook vectors using their weighted means and weighted covariance matrices (weighted by the "mass" of each codebook vector, i.e. the number of location samples associated with it).



(a) *Distribution of place counts over the size categories. (.95 CIs shown.)*

(b) *Distribution of place counts over the participants.*

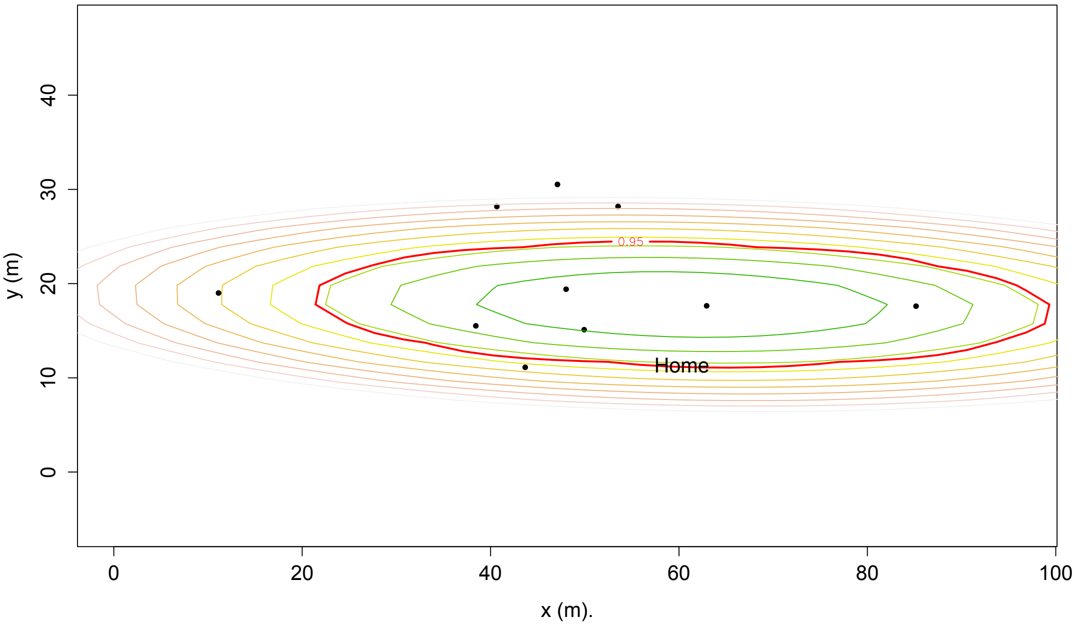
Figure 5.10: *Place size distributions*



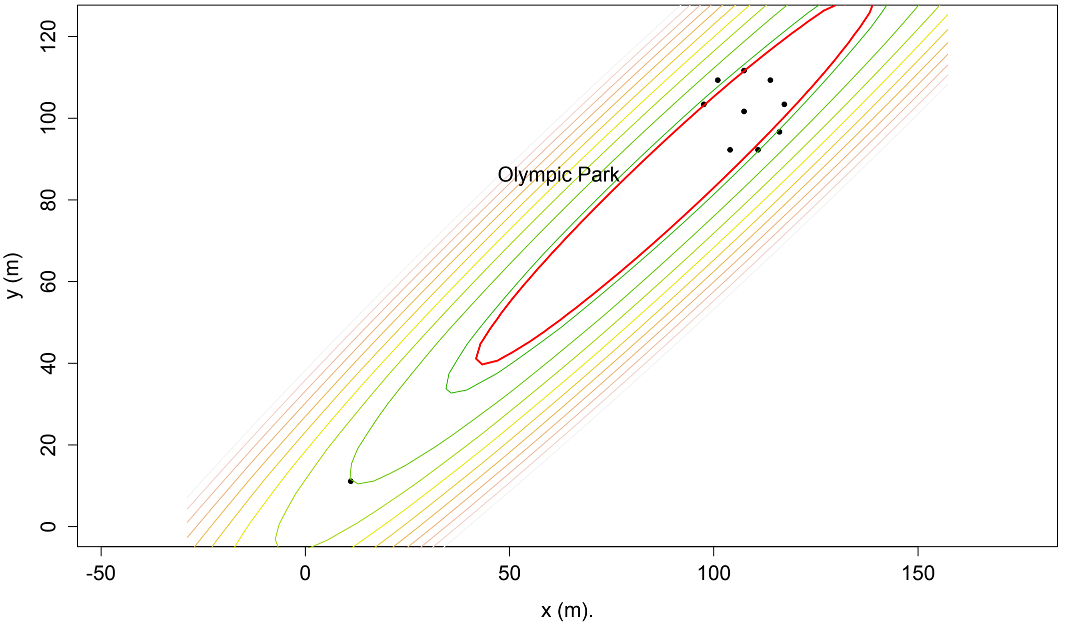
(a) *A selection of nearby places in Euclidean space for Participant 2. Each place's codebook vectors are shown with bivariate Gaussian density contours (0.95 quantiles are highlighted), parameterised by the weighted mean and weighted covariance matrix of the codebook vectors. A collection of ROOM and <ROOM sized places within a single building can be seen on the right. "Gym" is a BUILDING sized place.*

(b) *Two $\leq$ROOM sized places for Participant 1 that are $\approx 40m$ apart.*

Figure 5.11: *Visualising places from their codebook vectors.*

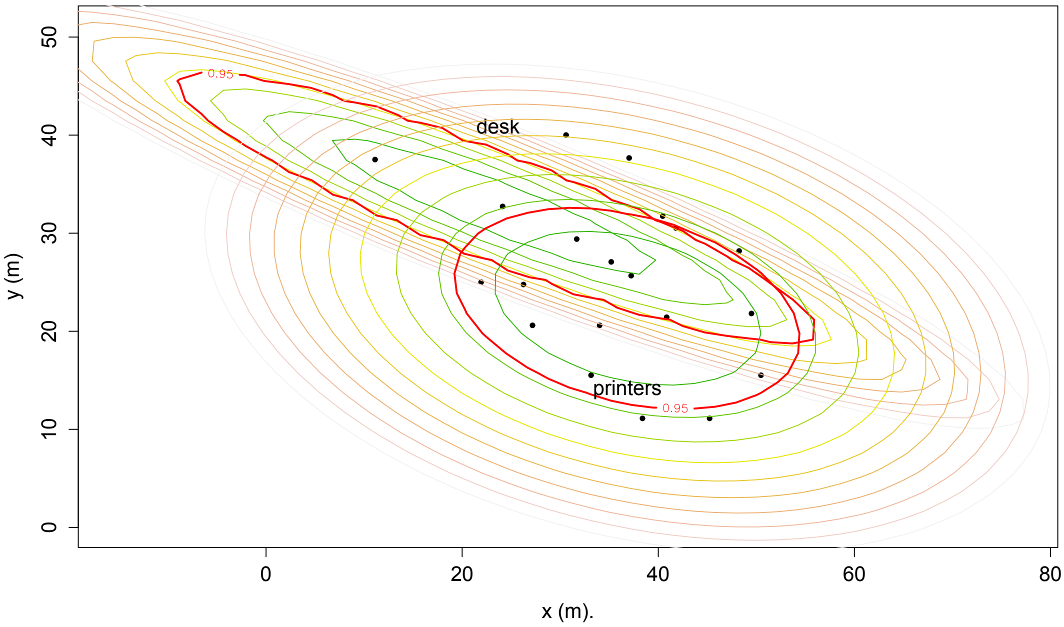


(a) *Participant 9's home, showing how updates from different rooms of a house have affected the place's codebook vectors and subsequent density.*

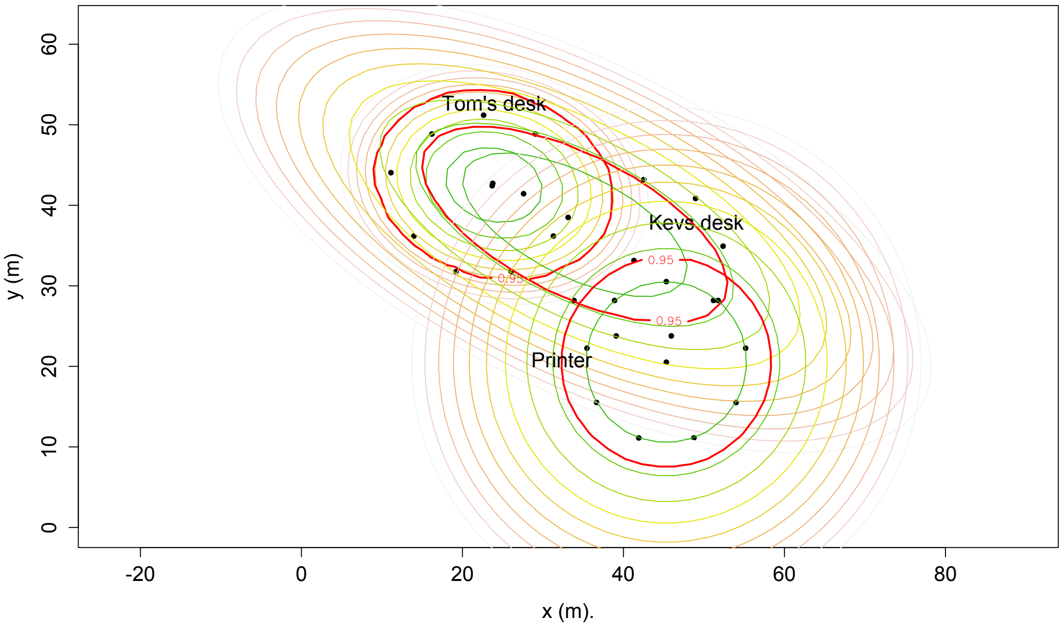


(b) *A >BUILDING sized place for Participant 6: the London 2012 Olympic stadium, showing how updates from different areas of a large place have affected the codebook vectors and subsequent place density.*

Figure 5.12: *Visualising large places.*



(a) *Places that are very close together within Participant 5's office*



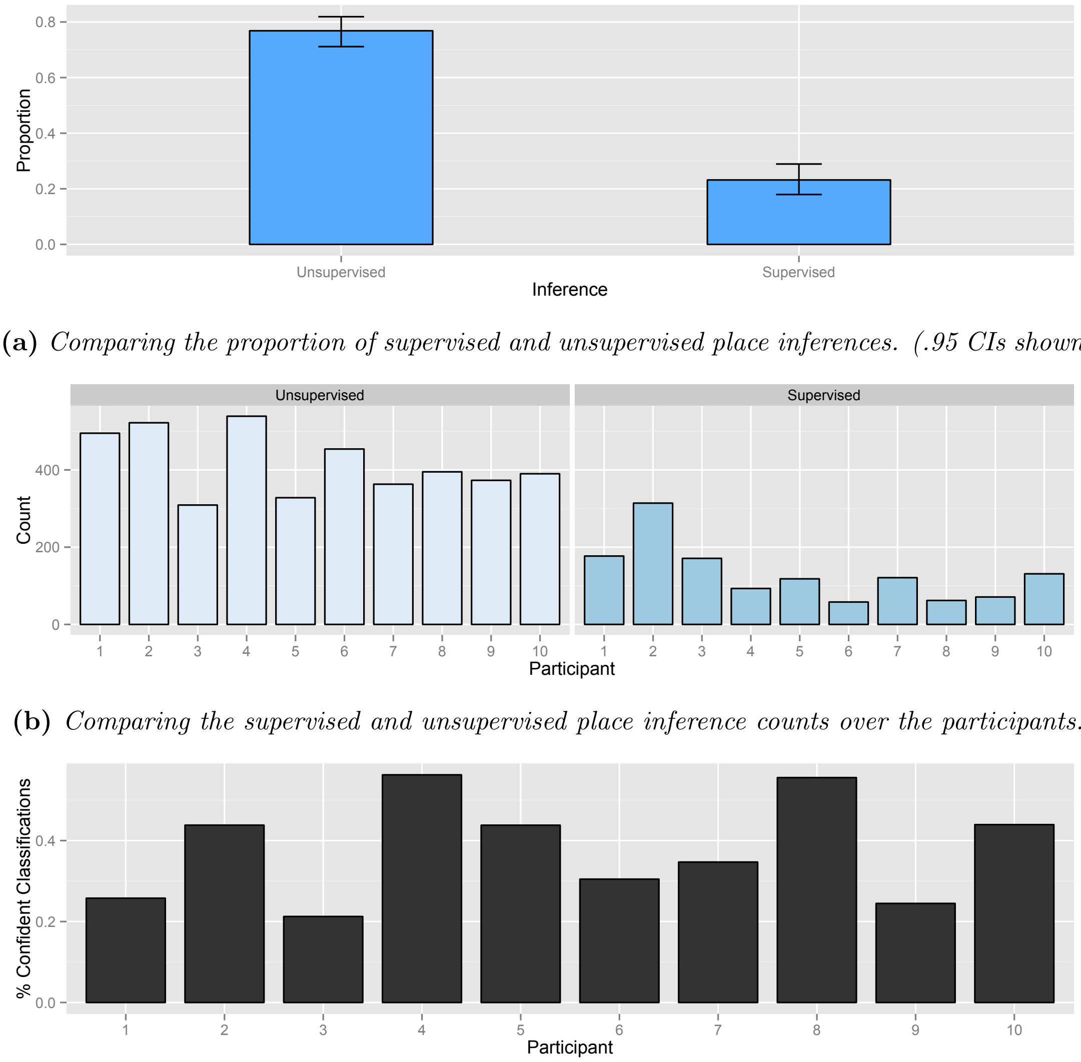
(b) *Multiple proximate office places for Participant 8.*

Figure 5.13: *Visualising small places within an office.*

Feedback Behaviour

Here we outline the user feedback results from the study. Figure 5.14 compares the supervised and unsupervised inferences during the field study. There was a significantly greater proportion of unsupervised inferences than supervised ones ($p < 0.05$; non-parametric bootstrap; 1000 replicates; $N = 10$).

Figure 5.14c shows the proportion of inferences that were confident during the field study, i.e. inferences in which the p-value of the Hotelling's T^2 confidence measure was greater than 0.05.

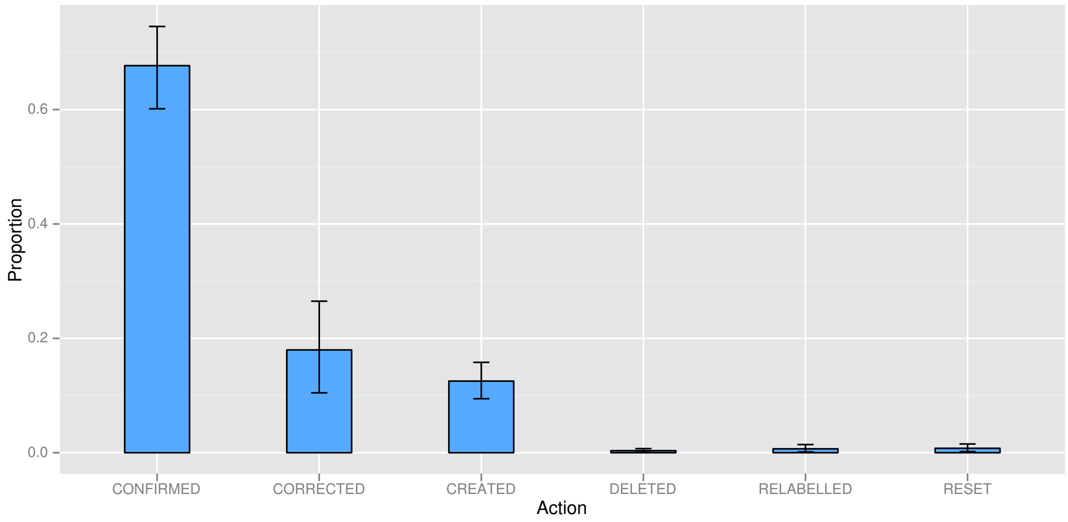


(a) *Comparing the proportion of supervised and unsupervised place inferences. (.95 CIs shown)*

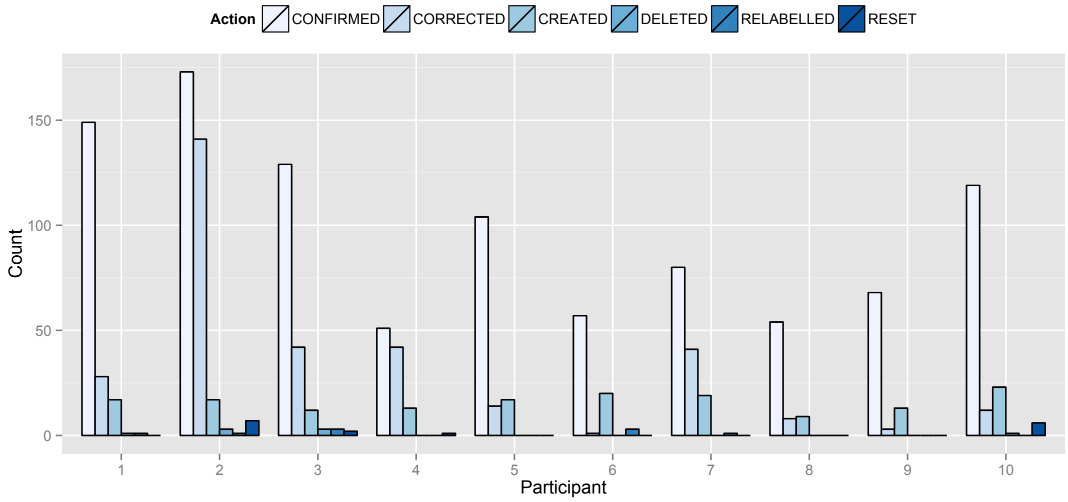
(b) *Comparing the supervised and unsupervised place inference counts over the participants.*

(c) *The percentage of place inferences that were confident over the participants.*

Figure 5.14: *Unsupervised and supervised place inferences.*



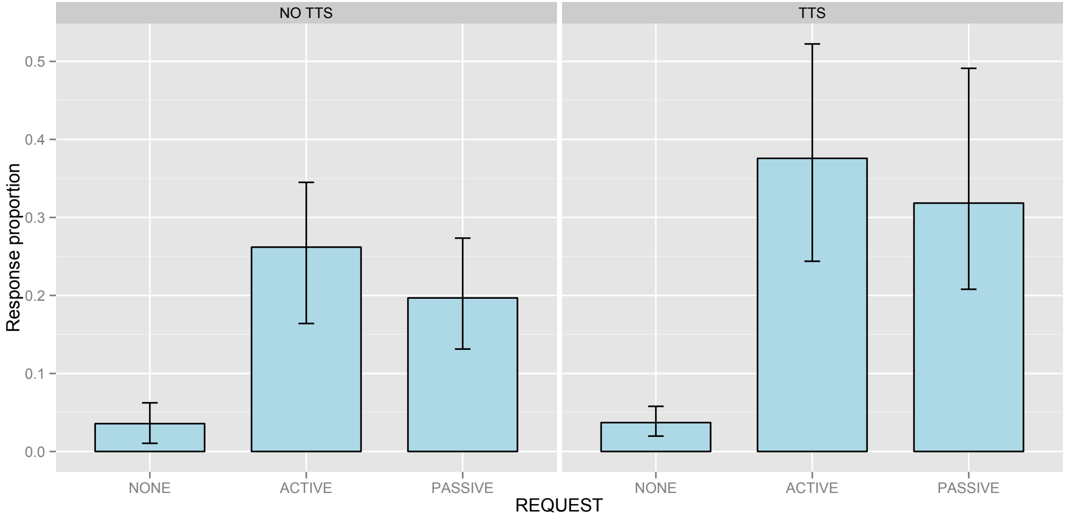
(a) *Distribution of feedback proportion over actions for the study. (.95 CIs shown.)*



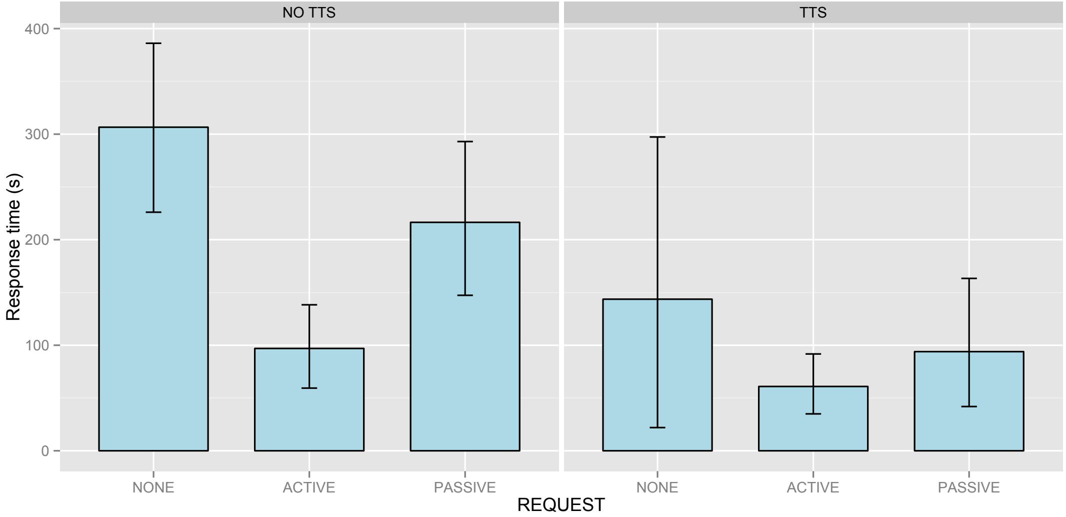
(b) *Distribution of feedback actions over the participants*

Figure 5.15: *User feedback action data.*

Figure 5.15 shows the distribution of user feedback actions, both over the type and the participants throughout the study. There are significantly more confirmations than any other action ($p < 0.05$; non-parametric bootstrap, 1000 replicates, $N = 10$).



(a) *The proportion of feedback requests responded to over the request and prompt conditions*



(b) *The median response time for supervised inferences over the request and prompt conditions*

Figure 5.16: *User feedback response results. The NONE condition shows the measures for un-requested feedback to allow comparison. (.95 CIs shown.)*

Figure 5.16 shows the user feedback response statistics. As Figure 5.16a shows, there is a small but insignificant increase in mean response rate between ACTIVE and PASSIVE requests within each prompt condition, and a small but insignificant increase in mean response rates for ACTIVE and PASSIVE requests between the prompt conditions ($p > 0.05$, non-parametric bootstrap, 1000 replicates, $N = 10$ in all cases). Figure 5.16a also shows the proportion of unprompted feedback, i.e. the proportion of confident inferences in which the user intervened.

Figure 5.16b shows a significant decrease in median supervised inference response time between ACTIVE and PASSIVE requests for the non-TTS prompt group ($p < 0.05$, non-parametric bootstrap, 1000 replicates, $N = 10$), and a small but insignificant decrease for the TTS prompt group. There is a small but insignificant decrease in response time for ACTIVE and PASSIVE requests for the TTS group compared to the NO TTS group ($p > 0.05$, non-parametric bootstrap, 1000 replicates, $N = 10$ in all cases). The median was chosen as a measure due to its outlier insensitivity and because the response times are not normally distributed, and Figure 5.17 shows the grouped histogram plots for the supervised inference response times over the request and prompt conditions. There is no significant correlation between the number of "Where are you?" messages received and the response rate or median response time for all conditions ($p > 0.05$ in all cases; Kendall's τ).

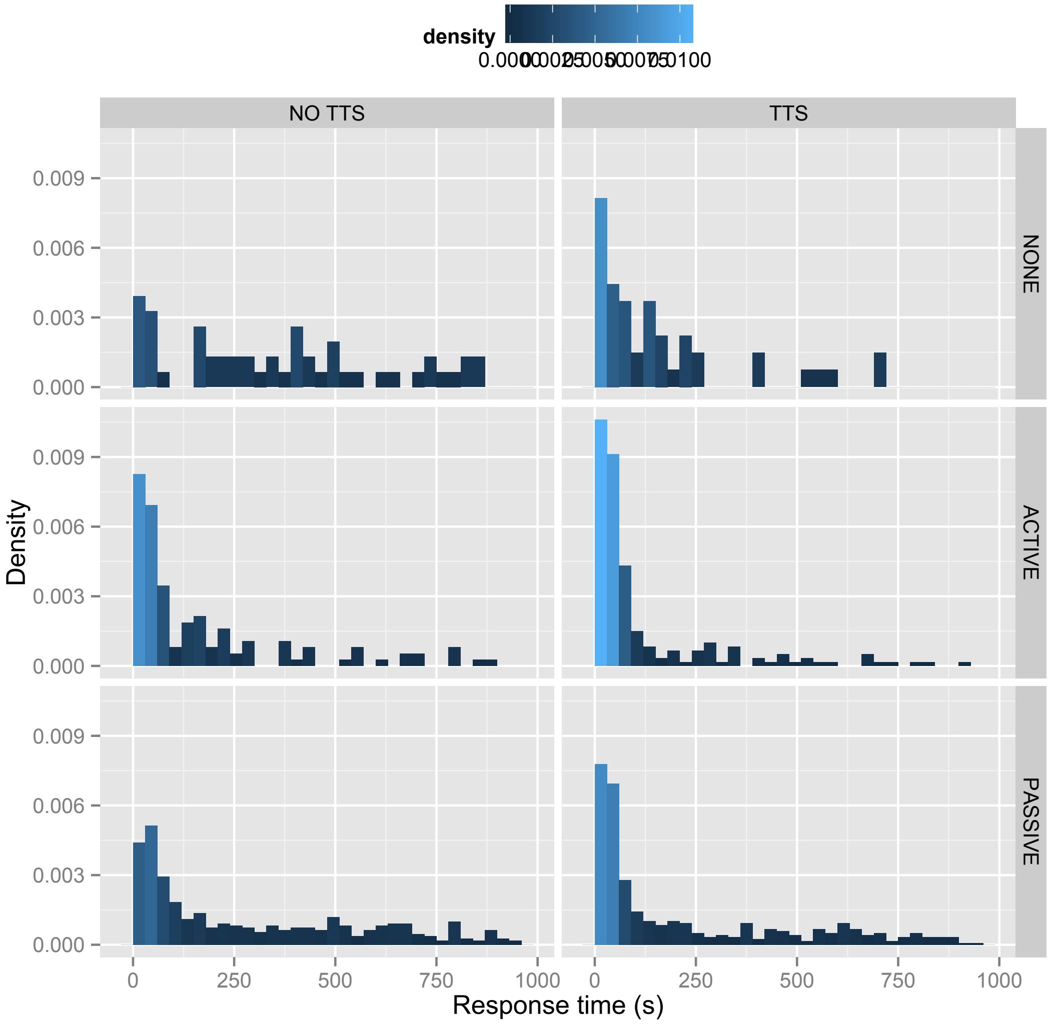
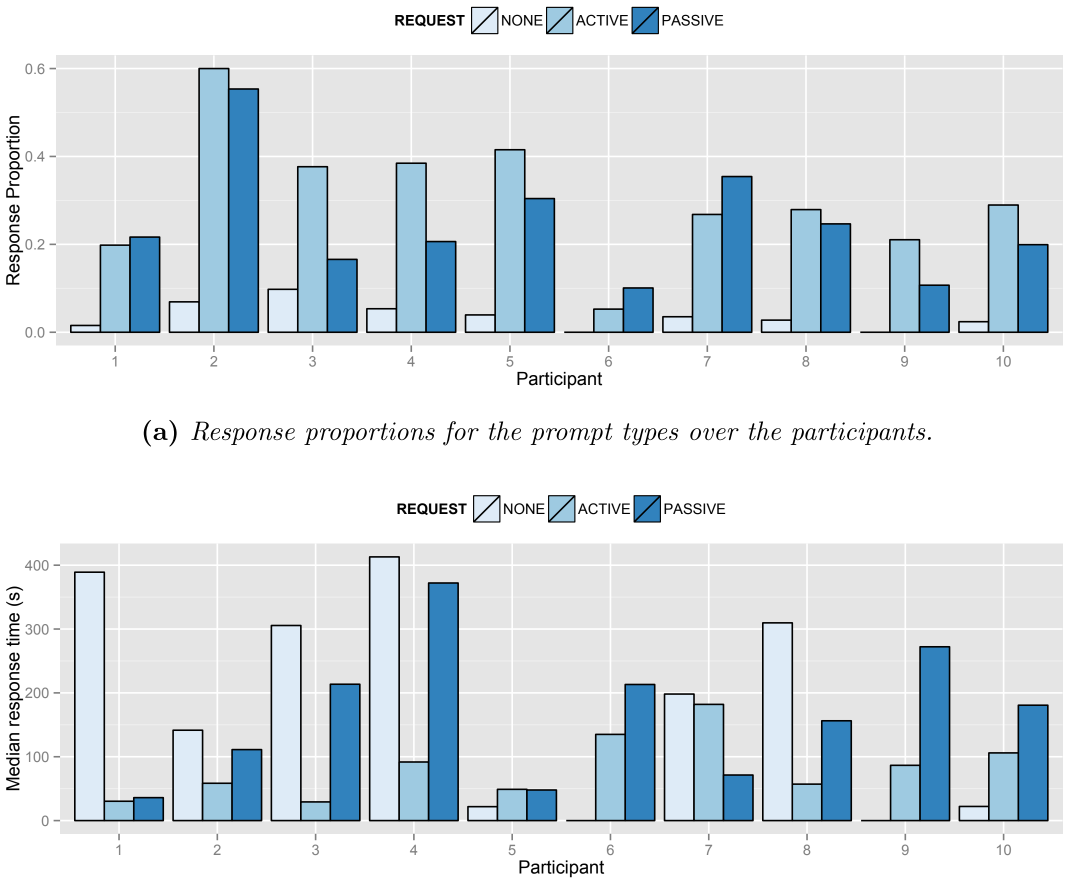


Figure 5.17: *Histograms of response times for supervised inferences over the request and prompt conditions*

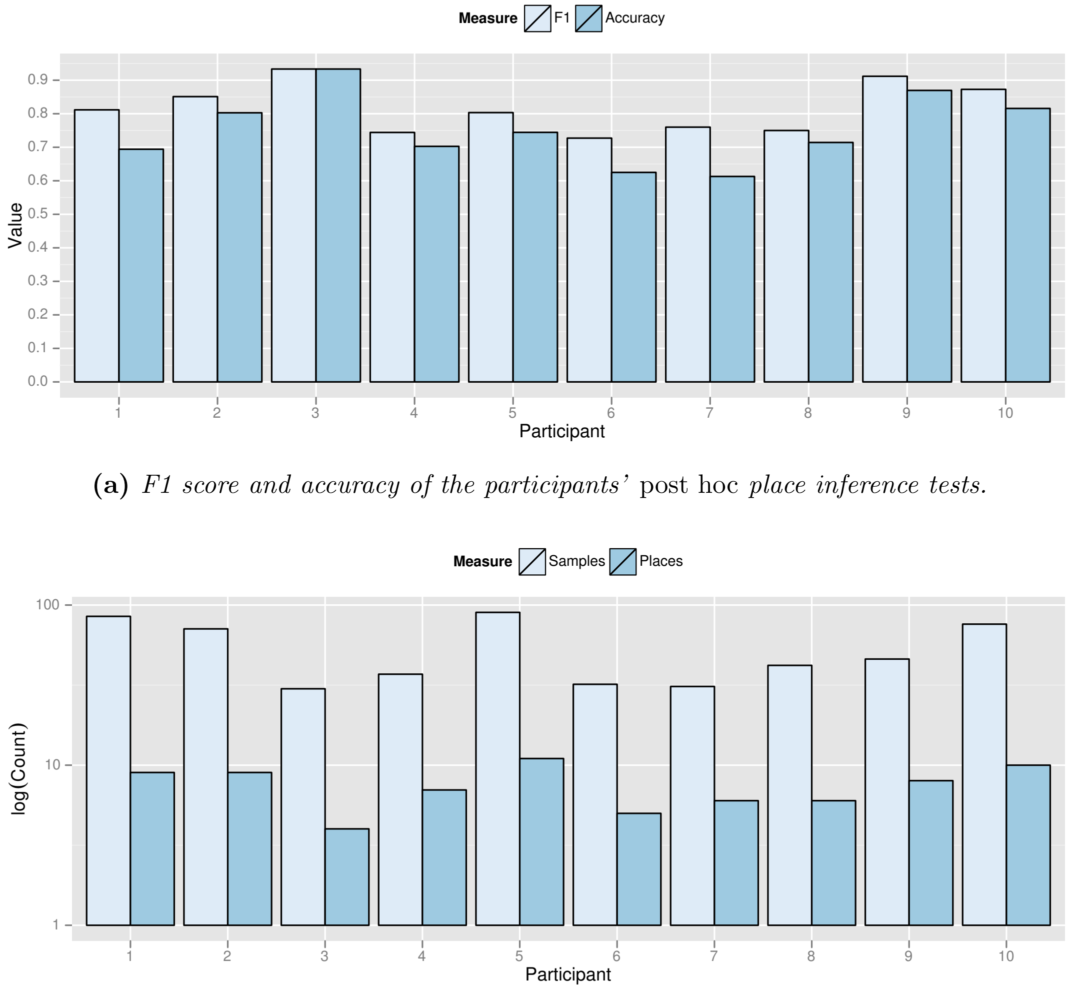


(a) *Response proportions for the prompt types over the participants.*

(b) *Median response times for the prompt types over the participants.*

Figure 5.18: *Participant distributions for request responses.*

Figure 5.18 shows the distribution of response rates and median response times over the study participants, grouped by request type.



(a) *F1 score and accuracy of the participants'* post hoc *place inference tests.*

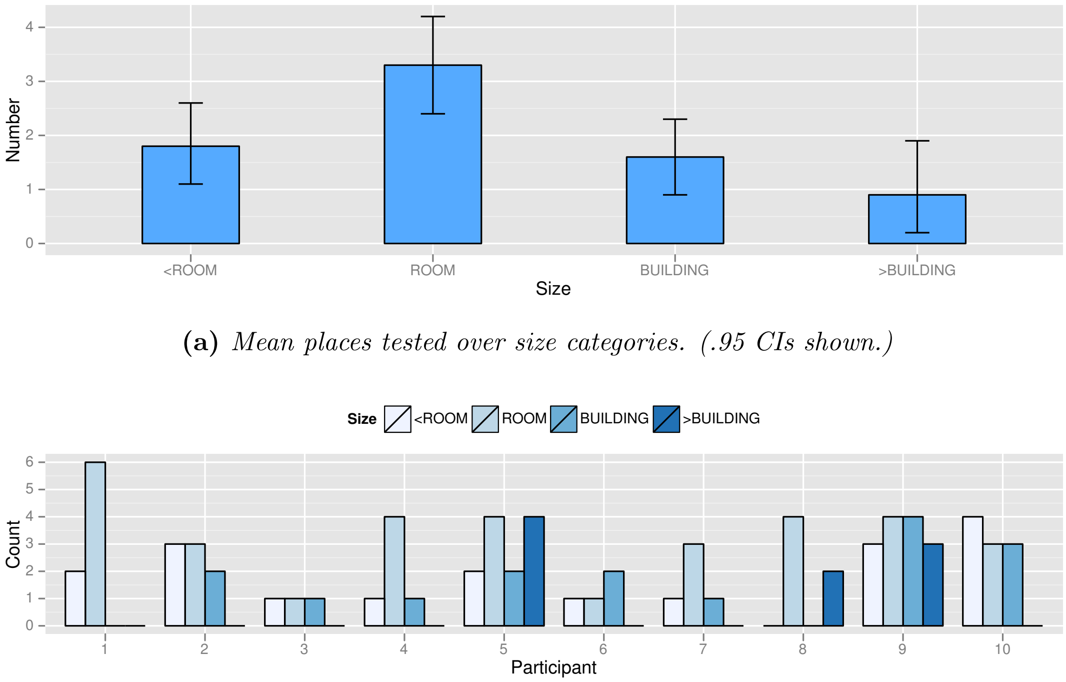
(b) *Test place and sample counts for the participants'* post hoc *place inference tests.*

Figure 5.19: Post hoc *place inference performance.*

Place Recognition Performance

Here we present the results of the *post hoc* place inference tests that were used to quantitatively measure the performance of the place inference process. Figure 5.19a shows the F1 score and accuracy results over the participants for their test runs. The mean F1 score is 0.82; 99% CI = $(0.75, 0.87)$ (non-parametric bootstrap, 1000 replicates, $N = 10$); sd= 0.07, with mean accuracy 0.75; 99% CI = $(0.67, 0.83)$ (non-parametric bootstrap, 1000 replicates, $N = 10$); sd= 0.1. There is no significant difference in F1 score or accuracy between the two prompt groups ($p > 0.05$, non-parametric bootstrap, 1000 replicates, $N = 10$).

Figure 5.19b shows the place and sample counts for the participants' test runs. The mean number of samples taken per participant was 54 (sd= 23.8) and the mean number of places tested was 7.5 (sd= 2.3).



(a) *Mean places tested over size categories. (.95 CIs shown.)*

(b) *Number of places tested over the participants.*

Figure 5.20: *Post-study test run performance*

The place size distributions for the participants' test runs are shown in Figure 5.20. Significantly fewer >BUILDING sized places were tested than ROOM sized places ($p < 0.05$; non-parametric bootstrap 1000 replicates, $N = 10$)

Table 5.7 shows the distance matrix in metres for Participant 1's test run. All participants' distance matrices can be found in Appendix C.3.

	Desk	Kitchen	Costa	Music.room	Claverton.rooms	ICIA	SU.upstairs	Balcony
Desk	0	160	220	340	400	300	220	0
Kitchen		0	360	500	540	440	360	40
Costa			0	140	180	80	0	360
Music room				0	60	80	140	500
Claverton rooms					0	100	180	540
ICIA						0	80	440
SU upstairs							0	340
Balcony								0

Table 5.7: *Distance matrix for Participant* 1*'s test places. Distances are in metres rounded to the nearest* 20*m*.

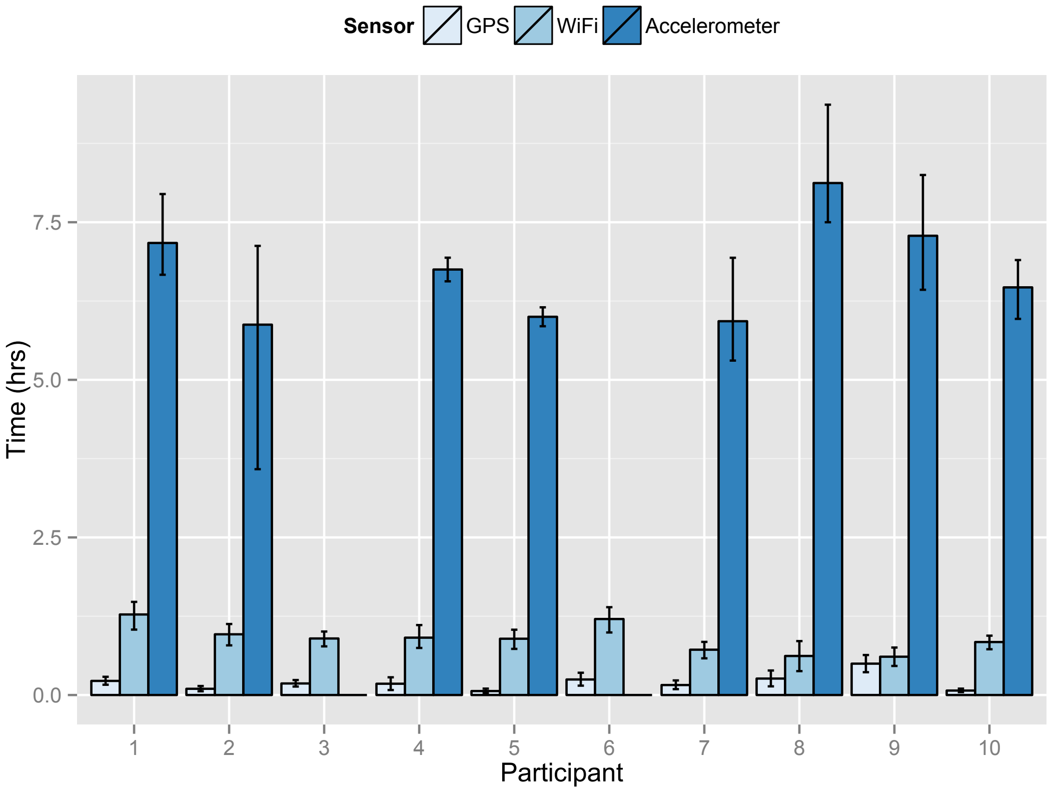
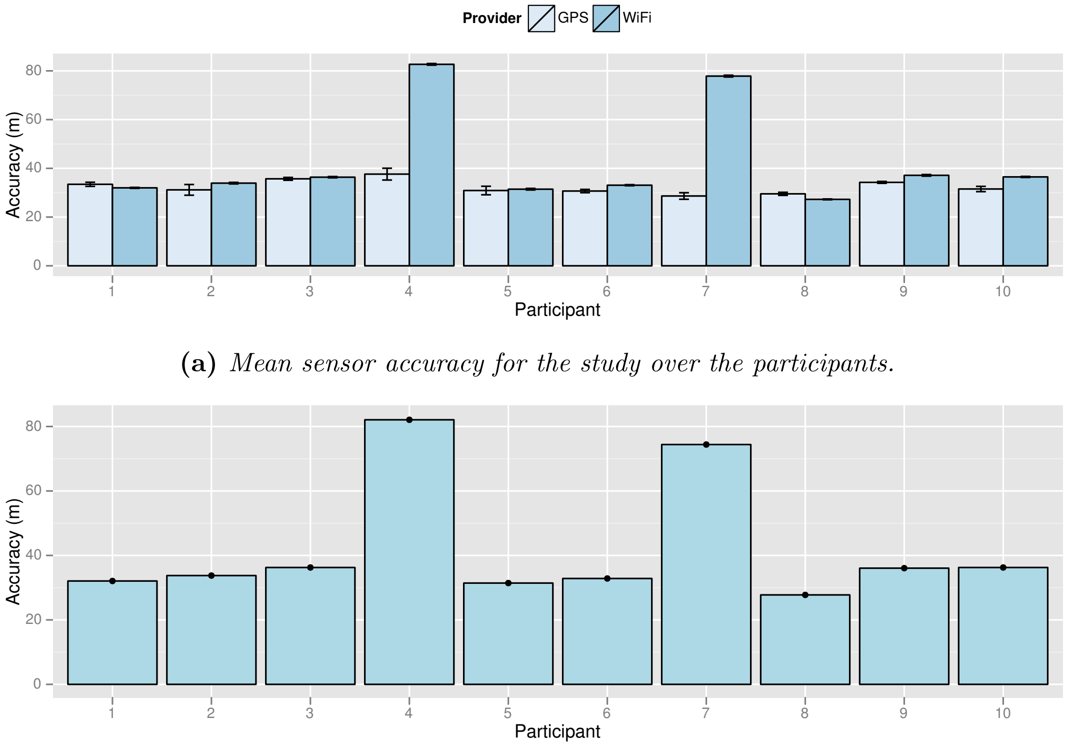


Figure 5.21: *Mean daily sensor usage over the participants throughout the duration of the study. Accelerometer data for participants* 3 *and* 6 *are not shown due to a technical issue with logging sleep data. (.95 CIs shown.)*

Resource Usage

Figure 5.21 shows the mean daily sensor usage for the participants throughout the study. The mean data storage per device for the two week field deployment was ≈ 600kB.



(a) *Mean sensor accuracy for the study over the participants.*

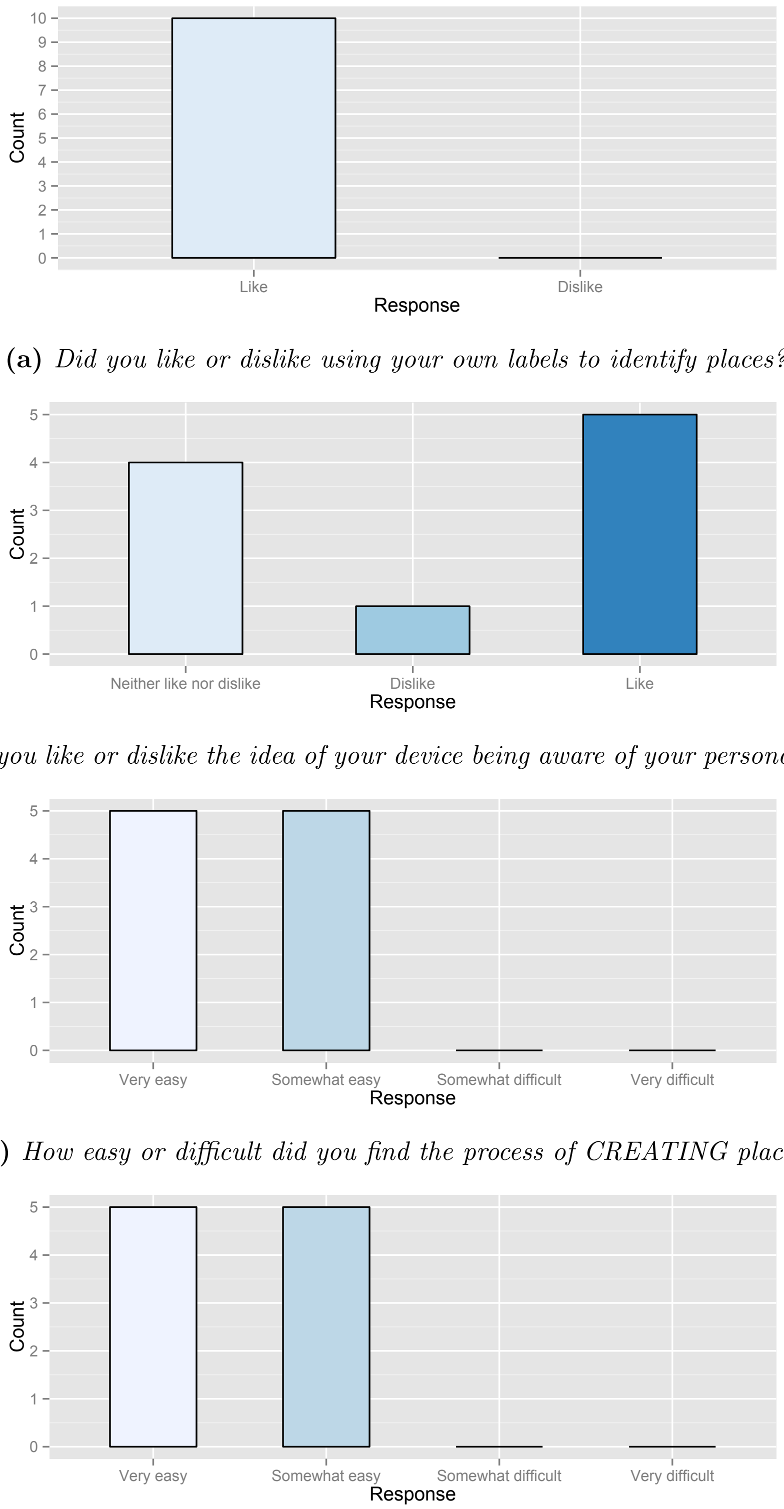
(b) *Mean overall location accuracy for the study over the participants.*

Figure 5.22: *Location accuracy data. (.95 CIs shown.)*

Figure 5.22 shows the summary statistics for location accuracy over the participants and sensors. The raw location accuracy data is detailed in Figure C.1 in Section C.4 of Appendix C.

User Experience

Figures 5.23 – 5.25 show the exit survey questions and participant responses to each question.



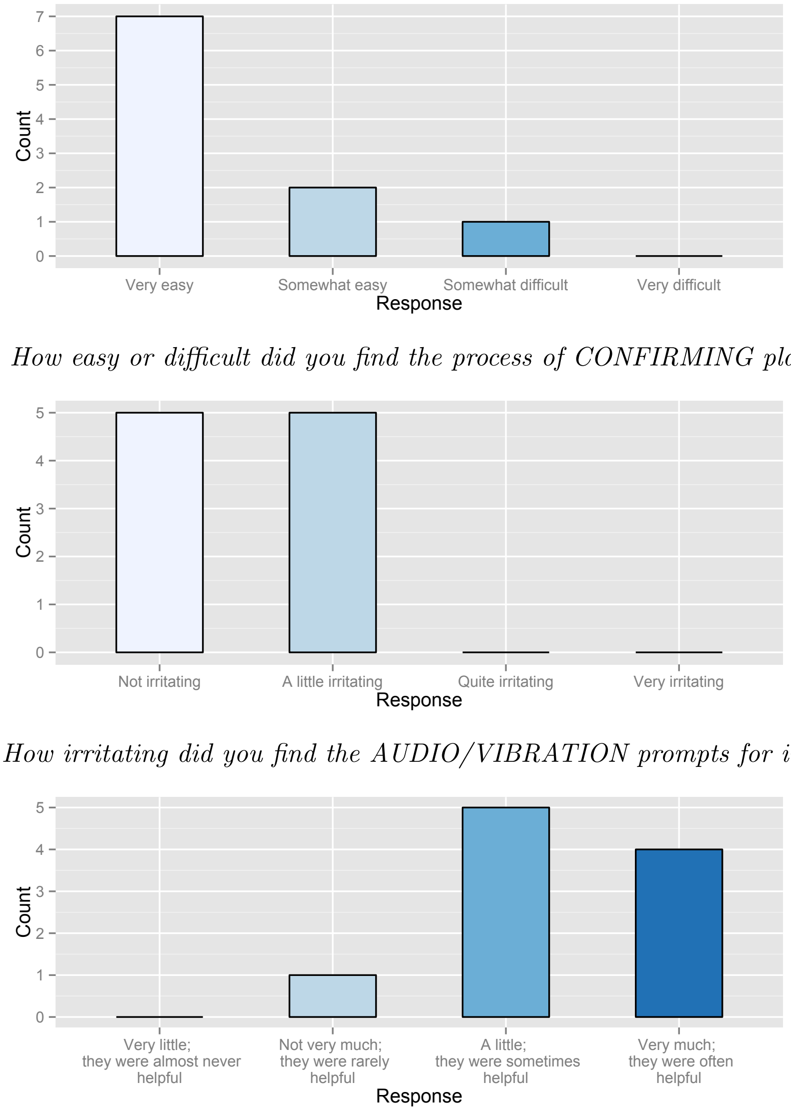
(a) *Did you like or dislike using your own labels to identify places?*

(b) *Do you like or dislike the idea of your device being aware of your personal places?*

(c) *How easy or difficult did you find the process of CREATING places?*

(d) *How easy or difficult did you find the process of CORRECTING places?*

Figure 5.23: *Survey Questions 1 – 4.*

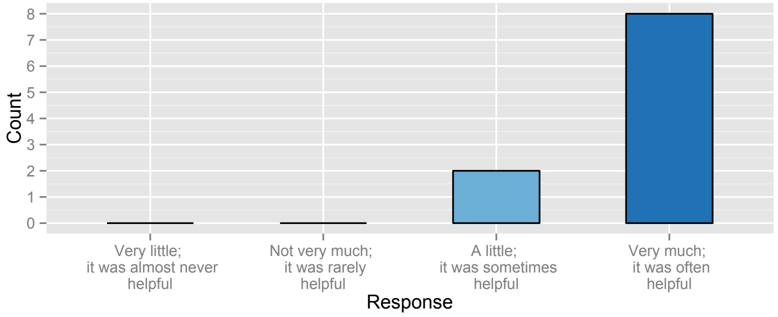


(a) *How easy or difficult did you find the process of CONFIRMING places?*

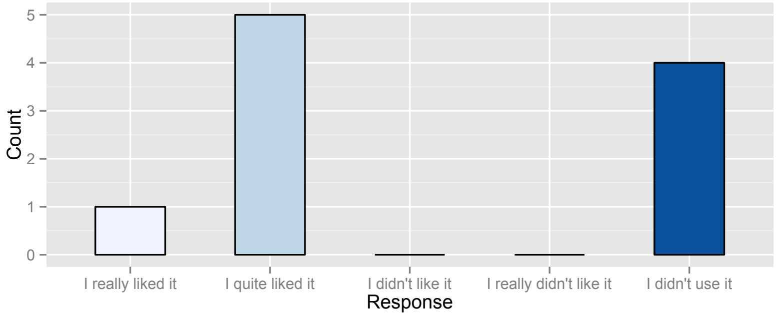
(b) *How irritating did you find the AUDIO/VIBRATION prompts for input?*

(c) *How much or how little did you feel the AUDIO/VIBRATION prompts for input affected the amount of input you actually provided?*

Figure 5.24: *Survey Questions 5 – 8.*



(a) *How much or how little did you feel the HOMESCREEN WIDGET affected the amount of input you actually provided?*



(b) *How much did you like or dislike the "Where are you?" application?*

Figure 5.25: *Survey Questions 9 – 10.*

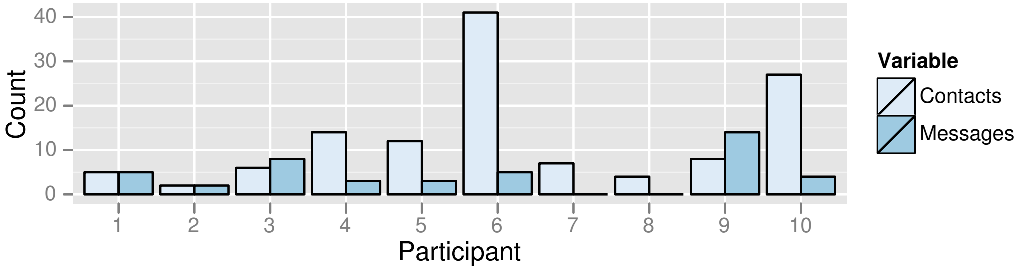
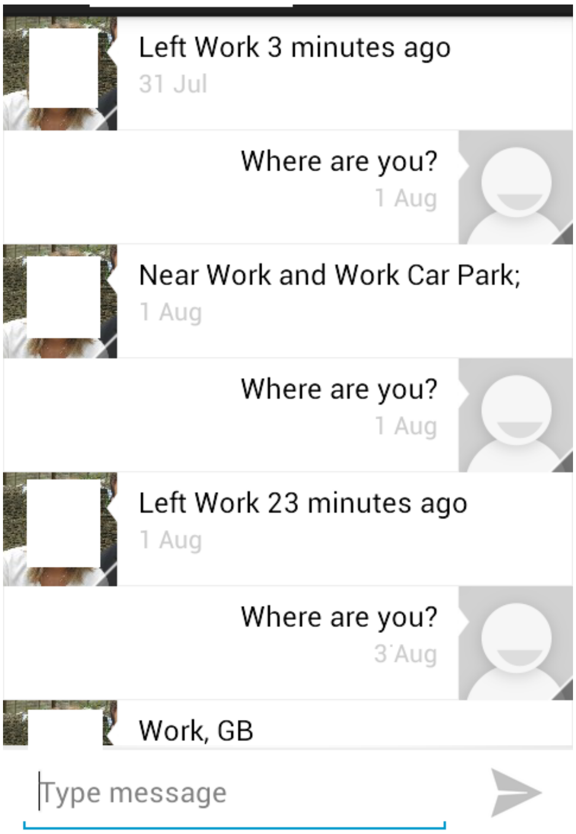


Figure 5.26: *Usage data from the "Where are you?" application.*

Figure 5.26 shows the usage data for the "Where are you?" application over the participants, and Figure 5.27 shows an example screenshot of the "Where are you?" application being used within the Android SMS application. An example of a low confidence inference – "Near Work and Work Car Park" – is shown, where nearby neighbours are listed (see lines 32–33 of Algorithm 8).



Left Work 3 minutes ago
31 Jul

Where are you?
1 Aug

Near Work and Work Car Park;
1 Aug

Where are you?
1 Aug

Left Work 23 minutes ago
1 Aug

Where are you?
3 Aug

Work, GB

Figure 5.27: *Screenshot from Android's SMS application showing a series of "Where are you?" SMSs and their auto-replies as the participant is: (i) walking between their work office and car park; (ii) driving home from work; and (iii) at work.*

5.2.7 Discussion

In this section, we discuss our results according to a set of key research questions. We also discuss the implications and limitations of the work, including lessons learned and possible extensions for future work.

RQ 4.1: To what extent can we infer and actively learn about users' meaningful places using mobile devices?

Without a report of what actually happened, it is difficult to speculate on the precise capture and learning efficacy of the place awareness system over the 2-week field deployment. However, with a median place count of 17 per participant, and a range from 9–23 places over all participants, it appears that the system captured many of the important places in each participant's daily lives. This is similar to the 2-week Connecto observational study [17] (6–20 places, mean 10), in which various methods to record what actually happened were employed. Moreover, during the performance testing, no participant identified a nearby 'missed' place, i.e. a place that was never captured by the system. Although the set of test places was smaller than the complete set of places captured, it does suggest that place capture was reasonably effective.

A particularly interesting finding is that the majority of captured places were approximately room-level size or smaller (Figures 5.10a and 5.10b), suggesting that users' place definitions are finer-grained and richer than other studies suggest, e.g. [115]. There is also a performance implication here – if many <ROOM sized places are in close proximity, it becomes extremely challenging to differentiate between them in real time without specialist location positioning hardware, cf. Participant 1's cluster of indoor places in Figure 5.11a. Kim *et al.* avoid this problem in [117] by considering places within a few seconds walk – that is, proximate room-level places – as a single place. This enhances the performance of their fingerprinting system, but it detracts somewhat from the idea that users should be able to define their own places without restrictions imposed upon them. Montoliu *et al.*'s approach [155] avoids the problem still further by using pre-defined place labels of places no smaller than building-level.

The learning effect was apparent in the dispersion of the field deployment codebook vectors in Figures 5.11 – 5.13, and their visual effect on the ellipses of the Gaussian density functions fitted to them. Further evidence of learning is reflected in the test performance results of Figure 5.19a and size distribution over the participants in Figure 5.20b, where participants' systems with a large range of tested place sizes (IDs 5 and 9) did not perform significantly worse than the mean F1 or accuracy scores.

The key result from the *post hoc* inference performance testing was that a mean F1 score of 0.82 and accuracy of 0.75 can be achieved on user-defined places in the field using this approach with only a small amount of user feedback. This is good performance considering the range of captured place sizes (Figure 5.10a), the high proportion of ≤ROOM sized places tested (Figure 5.20a), the mean location accuracy (Figure 5.22) and the absence of restrictions or assumptions placed on participants' places, e.g. size restrictions [117, 155, 133], naming restrictions [156, 155] and place variability restrictions [90, 116, 117, 115].

Figures 5.14 and 5.15 give an indication of performance in the field. Figure 5.15a in particular shows that participants were *confirming* significantly more inferences than they were *correcting*, suggesting that the many of the low confidence inferences were also correct. Moreover, as Figure 5.14a shows, there were significantly more unsupervised inferences (i.e. where users did not intervene) than there were supervised, suggesting that the device is performing the majority of place updates – $\approx 76\%$ – without any user feedback. Indeed, the device performed the majority of the place updates across all participants, as Figure 5.14b shows.

There are of course limitations when using a scripted tour approach to testing, particularly the range of places that can be tested with a high degree of observation precision. Indeed, participants could not test all of their meaningful places, and the majority of tested place sizes were ROOM sized rather than BUILDING sized (compare Figure 5.10a with Figure 5.20a); thus the test size distribution differs slightly from the field deployment size distribution.

Overall, place inference performance was good and required only a small amount of user feedback. Although feedback may be seen as a cost, it does provide various benefits: (i) it allows the capture of *in situ* place names and meaning through user-defined labels; (ii) it allows the capture and recognition of places from smaller than room level to greater than building level; (iii) it engages the user and provides intelligibility about the underlying system operation; and (iv) automated inference failures can be corrected with simple, low-burden user feedback.

RQ 5.1: How do different user feedback requests affect feedback response rate and time in a mobile environment?

There are a number of interesting findings from the results in Section 5.2.6. First, the participants only responded to ≈ 26% of ACTIVE requests and ≈ 20% of PASSIVE ones for the NO TTS prompt condition, and ≈ 38% of ACTIVE requests and ≈ 32% or PASSIVE ones for the TTS prompt condition (Figure 5.16a). This suggests that in the field, participants do not usually notice the prompts, or they choose to ignore them. It should be noted that ACTIVE requests are raised far more frequently than PASSIVE ones, which results in more missed requests. For example, short visits to multiple places in succession could result in multiple ACTIVE requests and, if the user does not respond within the short time window of the visit, the next request will override its predecessor. However, even with a low request response rate, the capture, learning and inference performance was still good, indicating that little user feedback is required to achieve good performance.

ACTIVE requests are responded to significantly faster than PASSIVE ones (Figure 5.16b) for the NO TTS prompt condition. This is not particularly surprising, but it does indicate that notifying users *on transition* is likely to elicit a response that is *relevant*, with Figure 5.17 showing that a large number of ACTIVE requests were responded to within 1 minute. This significant difference in request response time is not apparent in the TTS prompt group however.

Many participants felt that the haptic feedback (vibration) component of the ACTIVE requests was the most effective at capturing their attention, e.g. Participant 8 "*only really heard the sounds at [their] desk or meetings, but [they] felt the vibration a lot* *more*". Participant 7 commented that they "*thought [they were] getting a text at first, but [they] got used to it buzzing after a while*". This raises an important issue when considering the efficacy of ACTIVE requests – mobile device users are often used to requests from, for example, SMS, and this familiarity may affect their response behaviour in the field.

RQ 5.2: How do different audio prompts affect feedback response rate and time in a mobile environment?

Comparing the two prompt groups, there is weak evidence to suggest that using TTS audio prompts rather than simple audio prompts results in a better prompt response rate and shorter response time. However, the differences between the groups are not significant, so the results do not strongly suggest this. Therefore, although TTS prompts are a novel method for indicating inference uncertainty, the results suggest that TTS is not a useful feature to include when prompting for user feedback.

Participant 2 thought that "*they [the TTS prompts] were funny more than anything*" and even used them as an audio confirmation (even though they were designed to *elicit* confirmation): "*hearing the voice say the right place sometimes made me think, ah, I don't need to bother correcting it*". Participant 2 also indicated that they had deleted a place – "Loo" – out of social embarrassment in response to a TTS prompt.

RQ 5.3: How resource intensive is the interactive intelligent mobile context aware system?

The resource use of the place awareness system is reasonable. As Figure 5.21 shows, the resource-intensive GPS sensor need only be on for a few minutes per day, with WiFi on for approximately an hour. By including a sleep function in the device, and by duty cycling the accelerometer to 50%, the accelerometer is on for approximately 6–7 hours per day. Comparing this with the state of the art WiFi fingerprinting method [115, 117] – where GPS, WiFi and the accelerometer are activated for approximately 2h, 4h and 20h respectively – our approach shows a significant improvement in energy use. Other real time approaches, e.g. [155] do not report their sensor usage, so we cannot compare our results with theirs.

RQ 6.1: What is the user experience like?

The user experience appears to be mainly positive. Referring to the exit survey responses in Figures 5.23 – 5.25, all participants liked the idea of using their own labels to identify places (rather than pre-defined categories), and half of participants liked the idea of their device being 'place-aware'. However, the remaining half were either neutral (40 %) or negative (10%) about 'place-aware' devices. During the post-study interviews, some participants said they were put off by the privacy aspect of place awareness. Participant 6 said that she "*liked the idea of [their] phone automatically using [their] names and things, but [they] don't like the thought of it knowing and sharing [their] names*".

All participants found the CREATE and CORRECT feedback processes 'Somewhat easy' (50%) or 'Very easy' (50%) to execute, suggesting that the user burden associated with these functions is low. The majority (90%) of participants found the CONFIRM process 'Very easy' (70%) or 'Somewhat easy' (20%), suggesting that the user burden associated with CONFIRM is very low – especially pertinent given that CONFIRM was the most common feedback function (Figure 5.15a). Interestingly, a single participant rated the CONFIRM process as 'Somewhat difficult'; during the post-study personal interview, this participant – having identified themselves as the one who selected this option – explained that this was because they "*didn't like the ['Are you sure?'] pop-up; [they] didn't want to click another button to tell it it's right*".

Half of the participants found the ACTIVE requests 'A little irritating' and the remaining half found them 'Not irritating'. This would suggest that the ACTIVE requests are not excessively irritating, but they do cause some annoyance. Participant 8 – in the NO TTS group – commented that "*the beep was similar to [their] text beep, but [they] got annoyed thinking [they were] getting loads of texts*". The majority of participants felt that the ACTIVE requests helped them to provide input, with 40% rating them 'often helpful' and 50% 'sometimes helpful'.

All participants found the home screen widget helpful in enabling feedback, suggesting that colour coding the inference certainty has a positive impact. It should be noted, however, that participants were instructed to place the widget on their device's home screen as part of the study, thus it was very visible to them throughout. This finding does suggest that embedding a small widget such as this within applications would be favourable for eliciting user feedback.

3 participants (1–3) thought that a place hierarchy system would be useful. "*I'd like to have been able to say oh, this place is in this one. It would be great to fall back onto the higher one if it wasn't sure about the lower one*" (Participant 3). This is potentially a very useful design feature, as falling back on higher-level places if location observations were inaccurate could improve the user experience.

The "Where are you?" application was reasonably popular. As Figure 5.25 shows, the participants who used the application all responded positively with "I quite liked it" (50%) and "I really liked it" (10%). The remaining 40% said they did not use it beyond two test messages. As 5.26 shows, most participants chose a small range of close approved contacts with which to automatically share their place labels with. Of the participants that used the application, each received between 5 – 10 messages from approved contacts over the 2-week field deployment.

One feature of the application that participants liked was the ability to communicate presence without explicitly revealing location, and the use of personal names for places. Participant 3: "*My girlfriend liked that we used names we both knew; it was nice and personal. We both know where my desk is, so sending ['desk at work'] meant that she knew where I was, and anyone spying may not unless they knew me better*".

Although the application was not the focal point of the study, it was received positively by all participants who used it, therefore supporting the notion that place awareness between close contacts is a useful application for this work. The lack of correlation between the number of messages received and the request response rates and times suggests that receiving queries from close contacts neither encourages nor discourages users to respond to feedback requests.

RQ 6.2: How does interactive intelligence simulation compare with field behaviour?

Comparing the performance test results in Section 5.2.6 with the simulation results in Section 4.5.2, we see that performance is very similar – with mean F1 of 0.82 in the field and 0.88 in simulation. Furthermore, we see that the amount of automation in the field is similar to the simulation (0.76 field; 0.78 simulation), suggesting that the Bernoulli sampling model that we used in simulation is a useful feedback model when parameterised by the participants themselves. Care should be taken however, as the simulation data in Chapter 4 were collected from 6 participants, and the field study data from 10; therefore further verification with larger datasets would be beneficial to increase the validity of the simulation approach.

Nevertheless, the comparison lends credibility to the simulation approach, which raises the possibility of prototyping context aware system designs whilst initially avoiding the expensive efforts of collecting observations.

Field Study Implications and Limitations

The key implication from this study is that good place inference performance can be achieved with only a small amount of user feedback. Our system is more resource efficient than the state of the art, and there are no limitations on infrastructure, i.e. there is no requirement for specialist location-positioning hardware, or place definition, i.e. users may define places however they like. Places are malleable and are updated over time as users revisit them; thus there is no dependency on existing wireless infrastructure (as fingerprint-based methods are).

There are, however, a number of limitations to this study, and they should be considered in future design. First, many participants' places are room-sized or below, and location sensing technology is not yet accurate enough to reliably differentiate between small, proximate places. Although some were successfully classified in the field (Figure 5.11b), others were too close to be reliably classified in the long term (the cluster of small proximate places in Figure 5.11a).

Other limitations lie in resource usage. Although our system is more resource efficient than the state of the art WiFi fingerprinting system, there is still room for improvement. The key problem – certainly in the Android operating system – is accelerometer sampling (see power statistics in [115]). If the sampling process were made more efficient by handset manufacturers, then this limiting factor could be reduced and the resource efficiency further improved.

The final limitation is the size of test performance place set. The problems with collecting high-quality observation data remain and, although scripted tours allow for a representative sample of places to be taken, they do not represent the whole captured place set.

5.2.8 High-level Implications and Limitations

RQ 5 asks how we might elicit context feedback from users in a mobile environment. Our low-level research questions – RQs 5.1–5.3 – break this down into elicitation strategies and resource use. Our findings from the field study help to answer the high-level RQ 5 by showing how users provide feedback to prompts from mobile interactive intelligence in the field. The lack of difference in elicitation metrics between audio prompt groups and within subject prompt strategies suggest that users do provide feedback regardless of small differences in the prompt methods. We have also shown that users only respond to a small proportion of prompts – mainly due to the mobile environment and variable attention of mobile users compared to desktop ones – but this does not have an adverse effect on inference and learning performance.

The key limitations of the work lie in the range of prompt strategies. We only explored audio and visual media, but tactile prompting could be considered (we did use tactile prompts simultaneously with audio ones in our study, but all participants received the same treatment so we did not compare presence and absence of tactile prompts) in addition to further visual or audio approaches, e.g. flashing LEDs or earcons [75]. Other limitations lie in the nature of the feedback options presented to users. We had a range of actions, including create or confirm, but these are by no means exhaustive. An interesting agenda for future work would be to define a standard set of feedback actions for users of mobile IIS, or even IIS in general given the lack of requirements in the literature [219]. Thus, the key areas that remain in order to fully answer RQ 5 are the exploration of other feedback elicitation strategies (and media) in a mobile environment, as well as testing them in a variety of use cases beyond place awareness.

RQ 6 asks about users' behaviour when interacting with a mobile IIS. The two low-level RQs explored the user experience and comparisons between field and simulation behaviour. Our findings show that the user experience is perhaps not as bad as one might think for a system involving user prompting and burden. The most novel aspect of the work is the validation of simulation, as this strengthens the case for using simulation of user behaviour in IIS – particularly when reliable empirical data is difficult or laborious to collect. We have also shown that users do not interact very often with the mobile IIS, which is different to more traditional desktop IIS that hold users' attention. This is perhaps not surprising, but it does highlight the differences in user interaction between desktop and mobile IIS.

Work still needs to be done in order to fully answer RQ 6, however, including direct comparisons between desktop and mobile IIS. This work could also better inform the development of simulation models (ours was a simple stochastic Bernoulli model) and ranges of such models could be compared so that researchers could select ones most appropriate to their work in the future. Moreover, a richer analysis of user experience could be undertaken, particularly given the recent focus on user experience in both academia and industry. A more formal undertaking should be conducted that not only identifies key metrics by which to measure user experience in mobile IIS, but also to test a range of prompt strategies using these metrics in order to better design mobile IIS interfaces.

5.2.9 Study Conclusion and Future Work

In this field study of the mobile place awareness system, we have shown that good place inference performance can be achieved in the field with only a small amount of user feedback and with the use of fewer resources than current RF fingerprinting methods. Captured places are entirely user defined, with no restrictions places upon size or labels. Moreover, the place models are malleable, and can evolve over time as more location observations are associated with each place. Further findings show that, although users typically respond to the minority of requests for feedback, this does not appear to adversely affect performance. By requesting user feedback at the moment of transition between places, it appears that we can receive timely feedback with little burden on the user's part.

We have also shown that text-to-speech (TTS) prompting does not appear to affect feedback request response rates or times, but a home screen widget that displays both the device's current inference and confidence – as well as allowing users to quickly intervene with the inference – is a useful interface component for enabling user feedback on mobile devices. Although a home screen widget is a highly visible component, the findings associated with it have implications for application-embedded user interface components, e.g. enabling painless user feedback as a 'side task' within another application.

Direct extensions to this work include the development of a hierarchy system of place inference, where 'places within places' could be defined and classified according to the certainty of each inference. 3 participants in our study mentioned the potential usefulness of this feature, and its possible improvement to the user experience. Although this work has attempted to avoid restrictions upon user-defined places, a simple 'this in that' hierarchical rule may offer improvement with little drawback associated with applying restrictions to users' place definitions.

Further extensions lie in the investigation of different inference confidence measures. We used a probabilistic measure that was reasonably static, i.e. it relied on standard but arbitrarily defined probability thresholds to determine if a inference was confident or not. Other measures could be used, e.g. distance-based measures, and thresholds could vary dynamically over time or respond to user feedback frequency, e.g. increasing the leniency of the confidence intervals as user feedback frequency increases, and *vice versa*. Furthermore – as was explained in the studies – further work could investigate the impact on results of varying such thresholds and possible methods of learning them for large scale deployment.

5.3 General Discussion

In this section, we discuss the extent to which we have contributed to answers for RQ 5 and RQ 6, as well as discussing some of the high level implications and limitations of the work in this chapter.

5.3.1 RQ 5: How can we elicit context feedback from users in a mobile environment?

The performance and feedback behaviour results from our field study support the results from Chapter 4 and suggest that user feedback has a positive effect on inference performance. Through the feedback provided by feedback, the inference processes can actively learn about users' important context states and improve their performance beyond what might be achievable through fully automated processes.

We should be careful to highlight, however, that our studies in this and the preceding chapter concentrated on the concrete use case of place awareness and – although it is feasible to generalise our approach to other context facets such as activities – more work is needed to further answer this research question for other forms of context.

5.3.2 RQ 6: How do users interact with an interactive intelligent mobile context aware system?

Our work in this chapter has shown that requesting user feedback at the moment of context transition is a viable strategy for eliciting feedback in a mobile environment. Moreover, we have shown that using speech audio prompts may positively affect feedback request rates and response times when compared to simple audio prompts; though more work is needed to determine whether the effects are significant.

We have also shown that requesting feedback through a visible interface component (in our case, a homescreen widget) can help elicit feedback when users interact with their devices for other reasons. This further suggests that embedding feedback interface components into other mobile applications may be a viable approach to feedback elicitation in a mobile environment.

Further extensions to this work are needed to fully answer this question however. For example, studying the effects of 'earcons' as well as TTS prompts on user feedback response behaviour [75], or different input modalities for providing feedback, e.g. through voice or gesture.

5.3.3 Implications

The key output from this work is an approach to real time mobile context awareness, which uses users' own labels and user feedback to enhance its context inference performance. Although we have used the case study of mobile place awareness, our work can generalise to further context facets, e.g. activity or social context awareness. The work results in the following design recommendations for mobile place aware systems that utilise user feedback:

- Use inferred context transitions as triggers or events, e.g. for notifications, synchronisation or application updates. These are independent of the place inference components of our work, and they can be used for a large range of event-based applications. We used them for triggering sensors and inference algorithms, as well as prompting users when necessary. We have shown that they have a positive effect on inference performance, sensor 'on time' and the elicitation of user feedback in the field.
- Use a combination of on-transition and periodic feedback requests. For on-transition requests, using speech-based audio prompts may result in better feedback request responses.
- Use small embedded components within the device's user interface that provide visual indicators of context inference certainty. We used our home screen widgets to both indicate the certainty level of place inference, and to enable simple user feedback. Participants found this component useful, and it could certainly generalise as an feedback component in multiple applications beyond place awareness, e.g. activity recognition, on-device contacts management or social media management.

We have also better explored the "Where" context facet, particularly in relation to the 'place' vs 'space' conjecture [86]. The work in this and the previous chapter has helped in the understanding of context, and illustrates that even in a seemingly well-defined facet, there is more to be explored (cf. Schmidt *et al*'s argument that there is more to context than location [208]).

5.3.4 Limitations

Our findings from both the simulation study in Chapter 4 and the field study in this chapter clearly show that user feedback allows for good context capture, recognition and learning in mobile context aware systems. What is less clear, however, are the efficacy of methods used to elicit user feedback. In the place awareness field study, we have only analysed a small set of modes for user feedback, but other modes, e.g. voice or gesture input [241], could and should be considered in future research.

Furthermore, we have only analysed the performance of a place awareness system, which is chiefly concerned with the "Where", "Who" and "When" facets of context awareness. We have not considered the "What" in our study, i.e. user activity, but our approach can be generalised to other context facets and other use cases. For example, we could employ the same algorithms from Chapter 4 in $\mathbb{R}^N$ feature space (we are not constrained to $\mathbb{R}^2$ latitude longitude space) so that we can employ user feedback in the capture, recognition and learning of further context facets.

Additionally, as discussed in Chapter 4, we have not considered the "Why" facet, which contains more complex and harder to obtain data such as human emotion or intent. We have alluded to the capture of meaning in place awareness, and this – as we have said before – opens up the possibility of using active learning to transfer intent and emotion data from the user to the machine intelligence, but more in-depth study is needed to improve the understanding of the "Why" facet.

Other limitations lie in providing incentives for users. Although our studies have shown that only a small amount of user feedback is required for good inference performance, there is still the question of user burden during feedback. Without a tangible reward for their efforts, users may quickly tire of feedback even if machine learning techniques lessen the requirement for it over time. Eliciting feedback implicitly within applications may help to alleviate this problem, e.g. embedding small widgets into application user interfaces that allow users to quickly confirm their context.

5.4 Applications for this Work

The final part of this chapter reviews a list of potential applications for the work covered in Chapter 4 and this chapter.

Although the key contributions of this thesis have focused upon the lower layers of our layer model, we should always be aware of potential applications for our work. Here we outline a set of potential applications directly related to mobile context aware systems. We broadly describe these applications according to a set of high-level categories, in which mobile context aware applications are commonly designed for.

5.4.1 Presence and Availability Applications

Presence and availability applications are fundamentally designed to communicate users' whereabouts to others. These may include basic location sharing applications, e.g. Google Latitude[5] or Foursquare[6], or instant messaging services and presence awareness services, e.g. Microsoft's Office Communicator or Google Talk. The goal of these applications is to relieve the burden associated with managing communication between users, e.g. attempting to call someone and discovering that they are unavailable, by

[5] URL: `http://www.google.com/latitude` (Accessed 2012-10-28)

[6] URL: `https://foursquare.com/` (Accessed 2012-10-28)

communicating presence and availability over various channels.

Presence and availability applications are a key target application for context aware systems. Indeed, the "Where are you?" application used in our field deployment of the mobile place awareness system is a presence and availability application. By communicating user-defined place labels between a set of close friends, family or work colleagues, presence and availability can be shared without necessarily revealing exact locations, e.g. a husband is likely to know immediately where his wife is when shown her place label 'Desk'. This has further implications for encouraging privacy in location-based services, where awareness is communicated through shared knowledge of labels [179] rather than an absolute reference, i.e. latitude and longitude.

At the time of writing, Google have started to utilise high-level place awareness as part of the 'Google Now' service within their Android mobile operating system[7]. Here, users' days are partitioned into 'Home' and 'Work', and services such as traffic or timetable information are delivered to the user just before they depart from one to the other.

5.4.2 Recommendation Applications

One of the more popular applications for mobile context aware systems are recommender systems. Hundreds of applications exist that recommend films, books, restaurants, attractions, friends, business connections and music based upon sensed and inferred context [4]. The reason these applications are so popular lies in the economic value of connecting the right people with the right product: manufacturers and service providers want to sell their products and services, and consumers only want to buy what is relevant to them. Any service or application that can broker such a transaction, i.e. enabling both a sale and customer 'delight', is valuable; hence there is a substantial industry and a large body of research surrounding recommender systems.

Amazon's recommendation engine is a ubiquitous presence on its website[8], and Netflix famously ran a competition for researchers to design its film recommendation algorithm – the winner received $1M; Netflix received a world-class recommendation algorithm [24]. Google uses location to tailor its mobile search results, with the goal of increasing relevance, and its maps service can recommend nearby businesses and restaurants based on location. These are only a few notable examples, but they serve to illustrate the popularity of recommender systems.

Recommender systems are therefore an ideal application for mobile context awareness. Both content-based and collaborative [5] recommender systems can make use of context

[7] URL: `http://www.google.com/landing/now/` (Accessed 2012-10-28)

[8] URL: `http://www.amazon.com` (Accessed 2012-10-28)

labels and knowledge of users' real time context states. For example, user-defined meaningful context labels could be more valuable to a content-based algorithm than a location description or address, and collaborative algorithms could match similar users based on the similarity of context labels. By increasing the level of personalisation, recommendations could become more relevant to the end user.

5.4.3 Resource Saving Applications

Another popular objective for context aware applications is resource efficiency. By 'resources' we mean anything of value to the user that requires expenditure for functionality, e.g. money, time, mobile phone battery power, processing capability or data transfer. Particularly in the area of power saving on mobile devices, simple ideas such as turning off power-intensive features on a device overnight to the switching of low-consumption radio when in a period of low activity can have positive effects. Indeed, many 'task killer' applications exist that allow users themselves to free up device resources; thereby giving them control and an incentive for use.

5.5 Conclusion and Chapter Summary

In this chapter, we studied the feasibility of interactive intelligence in mobile context aware systems, a key component of which is user feedback, i.e. the user affecting automated inference by providing feedback directly to the inference processes through a mobile user interface. User feedback can be seen as a component of active machine learning [210], and it allows a human user – or 'oracle' – to be involved in key stages of an automated inference process. We investigated interactive intelligence in the concrete case of mobile place awareness, a currently popular research area within mobile context aware systems.

In Chapter 4 we developed a place inference system using simulated user feedback. This simulation approach allowed us to compare designs for a prototype place inference system that utilised user feedback. Simulations are limited in their reflection of reality however, and the purpose of this chapter was to design for interactive intelligence in the field, and to gather field-based observations through our case study of mobile place awareness.

We therefore built upon our work from Chapter 4 by defining a set of key user feedback requirements that were used to design and implement a mobile device interface for user feedback in the field. We also defined a set of user prompting mechanisms in order to compare their effects on the elicitation of user feedback, and implemented a small place aware presence and availability application that used SMS to communicate places between users.

We deployed our system in an observational field study, in which 10 participants used the system for a 2-week period. During this study, we gathered data to analyse the capture and learning of places, as well as the frequency and types of user feedback that occurred. Following the field deployment, each participant's place models were tested as classifiers using previously unseen test data.

The key findings from this study showed that good place capture, recognition and learning performance can be achieved with only a small amount of user feedback and reasonable resource usage. A large number of places of various sizes were captured across all participants, with a mean inference performance (F1 score) of 0.82. Moreover, the performance and feedback results from the field study appeared to support those from the simulations in Chapter 4. Other findings showed that prompting users actively, i.e. with audio and haptic feedback at the moment of transition into a place, typically elicits a faster feedback response than prompting them passively, i.e. with a silent visual update to their home screen, although no significant difference in the amount of feedback elicited between prompting approaches was found. There was also no significant difference in feedback response rate or time when using text-to-speech (TTS) prompts, in which place labels are 'said' by the device instead of using a default prompt alert.

Participant feedback showed that participants were generally positive to the idea of user feedback in automated place awareness. The majority of participants found the feedback processes simple and not overly burdensome, and almost all felt that both ACTIVE and PASSIVE prompts were helpful in encouraging user feedback. However, there was a mixed response to the idea of mobile devices being aware of personally meaningful places, and some participants were frustrated by the lack of hierarchical awareness and the communication of inference uncertainty. Those participants that used the "Where are you?" place awareness application appeared to like it, with many citing it as a useful augmentation to traditional location sharing applications.

The final part of this chapter reviewed a set of application categories in which our work could be usefully applied, with the "Where are you?" place awareness application serving as an example of a presence and availability application. In conclusion, this chapter addressed RQ 5 and RQ 6 through a field study of mobile place awareness with interactive intelligence. We showed that good inference performance can be achieved with only a small amount of user feedback – the majority of which is confirmatory, i.e. users are simply telling the system that its inferences are correct. We also showed how users respond to requests for feedback in the field using different request approaches and audio prompt approaches.

Chapter 6

Conclusion and Future Work

This dissertation has addressed a set of key issues related to the design and implementation of mobile context aware systems. In this chapter, we conclude the dissertation by summarising the key motivations behind our work, the scope of the work and the specific research questions that we addressed with the work. We then summarise our contributions and describe how they address the specified research questions, before analysing the work's implications for mobile context aware systems, as well as specific limitations of the studies that we undertook.

6.1 Dissertation Summary and Outcomes

The chief motivation of our work is the sheer popularity of mobile devices such as smartphones and tablets in everyday life. Their use by so many people means that there is opportunity to improve people's lives through these devices. The benefits of incorporating context awareness into mobile devices are becoming slowly realised outside academia; systems such as Google Now and Apple's Siri are good examples of this, as they demonstrate how context awareness can be used to lessen the burden of everyday tasks such as travelling and scheduling meetings with colleagues.

The key research problem that motivated our work is the lack of systematic design principles in mobile – and indeed general – context aware systems. The field of context awareness spans AI and HCI and, as such, it is a broad research area that is particularly fragmented, i.e. researchers typically concentrate on either the HCI or AI principles; few address the complex interactions between the two disciplines. This is the chief motivation behind the emerging field of interactive intelligent systems (IIS) [105], which studies the intersection between AI and HCI.

This dissertation has coupled these approaches together, provided a structure and addressed a set of practical issues associated with the implementation of mobile context aware systems, namely: the availability of context sources and sensors in a mobile environment; the difficulties of designing and implementing interactive intelligence on-device; and how users may be able to assist in context inference and learning process by providing feedback to the system.

The overarching question that our thesis addressed was:

- **$\mathbf{RQ_H}$**: How can we improve the design and implementation of mobile context aware systems?

Chapter 2 set the scene for the dissertation and reviewed key literature in the fields of context awareness and mobile context awareness. A range of active research areas in mobile context awareness were identified and used to derive a set of concrete research questions to address our high level research question. We approached these questions using a layer structure similar to ones so often seen in context aware systems (see Section 2.6.1). Using this structure, our concrete research questions are grouped as follows:

- Context sensing:

  1. **RQ 1**: What entities might we consider as virtual context sensors?
  2. **RQ 2**: To what extent does combining multiple context sensors affect sensing performance?

- Interactive intelligence: the intelligence:

  3. **RQ 3**: To what extent can we infer significant changes in context using mobile devices?
  4. **RQ 4**: To what extent can we infer and actively learn about context using mobile devices?

- Interactive intelligence: the interaction:

  5. **RQ 5**: How can we elicit context feedback from users in a mobile environment?
  6. **RQ 6**: How do users interact with an interactive intelligent mobile context aware system?

Chapter 3 – Context Sensing – addressed RQs 1 and 2 by describing the distinction between ‘physical’ and ‘virtual’ sensors of context. Physical sensors are typically hardware sensors that are designed for sensing a specific property, e.g. temperature or – more abstractly – location. Virtual sensors are typically software services or applications that are not designed for sensing, but can be used as such, e.g. social media or email applications. In Chapter 3, we considered the everyday calendar as a virtual context sensor and conducted a field study in an office to assess its fidelity performance. Our findings showed that the calendar is a poor context sensor due to the inherent 'noise' of reminders and so-called placeholder events, i.e. events that appear in the calendar but do not occur. For RQ 2, we fused a range of context sensors that are homogeneous to the calendar's data fields and showed that sensing performance can be significantly improved through this approach.

One of the key problems in context awareness is knowing *when* to sense and infer context; particularly in resource limited mobile devices. Chapter 4 – Interactive Intelligence: the Intelligence – addressed RQ 3 by considering context as discrete states in a finite state machine (FSM), with transitions between the states. By linking these significant transitions to mobile device motion (something considered by Ho and Intille for identifying activity transitions [92]), we considered the case study of identifying transitions between users' meaningful places with mobile devices. For this, we developed an approach to processing motion data and undertook a hybrid field/laboratory study of users carrying mobile devices through sequences of their everyday places. We systematically analysed how the parameters of our design affected transition inference performance, and showed that good, real time transition detection performance can be achieved.

We further addressed the design and implementation of the intelligence component of interactive intelligence in Chapter 4. We approached RQ 4 by developing a context inference and learning algorithm – triggered by context transitions – that incorporates user feedback into the inference and learning process using a branch of machine learning known as 'active learning' [210]. By continuing our case study of place awareness, we applied this algorithm to data collected during another hybrid field/laboratory study of users undertaking a week's worth of transitions through sequences of their meaningful places. Using a simulation of user feedback – seeded by survey data provided by participants during the study – we systematically analysed how place inference performance might vary in response to 'natural' user feedback in the field. We compared three alternative implementations of the algorithm, showing that a probabilistic approach to place classification – rather than a geometric approach – appears to be the superior design choice.

In Chapter 5, we considered the interaction component of interactive intelligence in mobile context aware systems. Here we addressed RQs 5 and 6 by implementing our place awareness system on mobile devices in the field. Through a 2 week field study with a *post hoc* performance test, we showed that good place awareness performance can be achieved with comparatively little user feedback – a finding which supported the results of our simulation in Chapter 4 and contributes to answering RQs 4 and 5. We also designed a set of user interfaces to allow users to provide feedback in our mobile place awareness system, and compared alternate request types and modes, including: requesting feedback *actively* at the moment of transition vs requesting it *passively* when they interacted with their devices; and, for active requests, comparing how a simple audio prompt compared to a speech prompt (where the user's own place label was spoken to them at the point of transition). Our findings from the field study showed that: active requests significantly improve user response time for the simple audio condition, but no significant difference was observed for the speech condition. There was also no significant difference in response rate for either prompt condition or request type. We also showed that our approach used fewer sensing resources than the current state of the art, and that the user experience was generally positive.

6.2 General Implications

There are several implications of the work in this dissertation. First, in reference to our high level research question, the work can better inform the design and implementation of mobile context aware systems. By structuring a mobile context aware system using a layered approach at the design stage, designers can concentrate on developing each of the layers independently of the others; agreeing only upon the interfaces between them. At the implementation stage, developers can ask specific questions relating to each layer, e.g. what sources and sensors can we use to satisfy our application requirements, given our potential constraints? What inference and learning methods can we use? Do we involve users in the inference process? If so, how? Second, our work considers interactive intelligence in mobile context aware systems. Although much of the existing work in mobile context awareness focuses on either intelligence, i.e. context inference and learning, or user interaction, the intersection between the two has largely been neglected; as it has until recently in desktop systems [219, 218, 220]. It is an important area however, as intelligence should be designed with interaction in mind and *vice versa*. This is being somewhat addressed by the interactive intelligent systems (IIS) community [105] due to its important implications for both intelligence performance and user experience. Our work has implications for both the intelligence and interaction components of mobile IISs.

The concrete implications of our work are as follows:

- For **context sensing**, designers should consider the fusion of multiple sensors in order to improve sensing performance. We showed how data fusion of the calendar with other context data can improve sensing, and this has implications for mobile context aware systems that connect to calendar systems. For example, traffic monitoring systems can use the calendar with other context sensors, e.g. location, to give predictive traffic updates with more confidence than simply reading the calendar naïvely.

- For **context inference and learning**, designers should consider using context transitions as triggers for more sophisticated inference processes that may also involve expensive sensing processes. We showed how mobile device motion can be used to infer transitions between users' meaningful places, which has implications for event-based triggers, e.g. triggering message synchronisation or notification prompting at the moment of transition into a new place. Designers should also consider involving the user in the inference and learning process by allowing the user to give feedback, in real time, on the inferences. Not only does this increase inference accuracy but, through the use of active machine learning techniques, users need not provide large amounts of feedback to ensure good performance. Indeed, as we have also shown through our mobile place awareness field deployment, users are unlikely to respond to the majority of requests for assistance but, as we have also shown, active learning allows inference processes to be robust to this.

- For **user interaction**, designers should – if user feedback can be incorporated into the context inference and learning process– raise requests *actively* at the moment of context transition using simple audio prompts (rather than speech prompts) in addition to tactile and visual prompts. We have shown how this results in faster prompt response times (though there appears to be no increase in response rate), which allows feedback to be provided in real time whilst the inference and learning processes are also executing in real time.

6.3 General Limitations

There are of course limitations to the work in this dissertation, and we summarise them here as caveats to the aforementioned implications. These are wider limitations than the individual limitations of each study (which are summarised in their relevant chapters).

The first key limitation of this work relates to the collection of real world observation data. We chose a range of methods according to each study, but collection remains a challenge in research [22, 52, 156, 195]. For the calendar study in Chapter 3, we used a combination of ethnography – observing participants in their natural environment – and self report with diary studies. Even with these approaches, we could not observe reality to as fine a degree of granularity as we would have liked. For our studies on context transitions and user feedback effects upon context inference in Chapter 4, we used scripted tours of participants' everyday place transition sequences for reliability purposes – a common approach for this type of study, e.g. [117]. Scripted tours are a hybrid approach with both laboratory and field components [195] that, although they allow reliable capture of reasonably valid data, compromise on pure ecological validity, i.e. the fully natural environment of a field study [156]. The simulation results are further validated from the *post hoc* performance results from our field study in Chapter 5, but these were also obtained using scripted tours (albeit without researcher accompaniment), and methodology limitations should still be noted when interpreting the results.

Another limitation relates to the generality of this work. For our studies, we necessarily had to choose case studies for evaluating our designs which, although they provide good data for analysis, do limit the generality of the results. One solution to this is to evaluate designs in multiple case studies, something which we will discuss further in the next section. Other generality issues to consider relate to our demographics, e.g. office workers in the calendar study, or the student/office worker demographics in subsequent studies, and also sample sizes. The laborious nature of capturing reality in our studies meant that we could not collect data from large samples of participants. Although similar to many other sample sizes in other UbiComp studies, e.g. [17, 50, 90, 117], we are still aware that larger sample sizes would increase the validity of the work. Further generality issues relate to mobile device types used in the studies: due to development constraints, we were only able to use a single device type or operating system, e.g. Android. Where possible, we tried to chose our participants based on their everyday use of particular device (usually by using their own, e.g. the field study in Chapter 5) but, again, we are aware that this restricted range may somewhat limit the validity of our work.

6.4 High and Low Level Research Questions

Here we discuss the implications and limitations of taking reductionist approach to the work in this dissertation. We began the dissertation by outlining a set of high-level research questions. For each chapter, we broke these down into sets of low-level questions in order to guide the tangible outputs of the research. There are of course implications and limitations to this approach, particularly in relation to the general impact and completeness in answering the original high-level questions.

In each of the study and chapter discussions, we have outlined the direct implications and limitations of the respective findings in the context of the higher level research question for which the work was carried out. In each case we discussed the extensive depth of the findings, but also their lack of breadth. We have partially answered each of the high-level questions, but there is still scope for further, broader work in order to better answer them in the long term.

In Chapter 3, we showed how our discovery that the calendar is a bad yet viable virtual context sensor contributes to the high-level research question RQ 1, which asks what might be considered a virtual sensor of context. The key limitation here lies in the lack of similar study for other virtual sensors, e.g. email or social media. The next stage to better address RQ 1 is to repeat our calendar study for a number of candidate virtual sensors, particularly those listed in Figure 3.3. A long term objective might be to formally catalogue virtual context sensors, along with a measure of quality relating to each context facet.

Also in Chapter 3, we also showed how fusing multiple context data improved the overall sensing performance compared to the calendar alone. This shows how a particular set of sensors behaves when combined, which is the purpose of RQ 2. If this could be repeated using different combinations of context sensors, as well as using different approaches to data fusion, then the breadth of knowledge for RQ 2 could be increased beyond what our findings have contributed.

In Chapter 4, we explored how well place transitions could be inferred using mobile devices. Place transitions are an example of a more general context change, and RQ 3 asks how well such changes might be inferred using mobile devices. Our study used motion to infer place change, and previous work has used motion to infer activity change [92], but other transitions remain to be studied. To further contribute to RQ 3, other context changes need to be defined and sensor data utilised to determine the extent to which general changes in context can be inferred.

In the latter half of Chapter 4, we explored RQ 4, which asks the extent to which we can infer and actively learn about context using mobile devices. We studied the case of place inference and active learning, and showed how inference performance is improved through active learning. The chief implication of our findings relates to the 'Why' facet of context: active learning could be used as a means of knowledge transfer between the user and the intelligence for more complex data such as emotion or intent. Rather than attempting to directly sense this data, which is demonstrably non-trivial [186], active learning could be used to elicit it instead. Furthermore, using active learning in data collection methods such as Experience Sampling [50] could 'offload' the burden of collection from the user to the intelligence, thereby retaining high accuracy with lower effort cost. There are areas that still need addressing in order to completely answer RQ

4, however. Namely, looking at other use cases beyond place awareness, and comparing further methods of active learning beyond the set that we explored [210].

In Chapter 5, we studied our mobile IIS – a place awareness system – in the field. Here, we were concerned with RQs 5 and 6, which ask about user feedback elicitation strategies and interaction in mobile IIS. Our findings showed that users are willing to provide feedback, but the proportion of prompts responded to is low. However, this did not adversely affect inference performance as intelligence is capable of functioning in the face of low user feedback rates. We also showed that there is little difference in elicitation strategies in a mobile environment and that the user experience is favourable. These findings help to answer RQs 5 and 6 but again, there are further avenues of research that are still required in order to fully answer them. Firstly, a range of mobile IIS and users' interactions with them should be studied in the field. This would increase the breadth of mobile IIS knowledge and better inform designers of likely user behaviour in relation to mobile IIS. Secondly, a standard set of feedback actions should be produced. We discussed the lack of such actions in Chapter 5 [219]. Finally, further output media and prompting strategies should be explored. We only considered visual and audio media, but additional knowledge could be gained by studying tactile prompts, or other forms of visual and audio prompts such as earcons [75].

In summary, our reductionist approach has yielded findings through deep exploration with narrow scope. The limitation of this is therefore breadth and, to fully address the high-level RQs, more breadth of work, rather than depth, is required.

6.5 From Layer Model to Architecture

In Chapter 2 and throughout this dissertation, we have used a layer model to guide and structure the research in a modular fashion. We raised the question of whether the layer model should be developed into a more formal software architecture for future developers of mobile context aware systems. Extending the model from its current state into such an architecture is certainly grounds for future work, but we have shown through its application in our studies that it is useful as an abstract representation of a mobile context aware system, and that it can be used to develop a fully functioning system with formal APIs (in particular, the place awareness system in Chapter 5). The first steps toward this might be to formalise and document a standard set of APIs between the layers, and develop the underlying functionality of the lower layers into libraries for various languages such as Java (for Android), Objective C (for iOS) and C/C++ for native operation on mobile devices and other embedded systems. A template for an API that could be engineered from the layer model is shown in Table 6.1. Here, each layer is a black box module with input and output methods based

I/O	Method	Description
	Sensing Layer	
Output	Who What Where When Why	Raw context data from individual or fused sensors.
	Intelligence Layer	
Input	(Sensing layer outputs) Create Confirm Correct Delete Reset Relabel	Context inference feedback from interaction layer, i.e. actions in Chapter 5
Output	Who What Where When Why	Inferred context using richer data structures, e.g. a ‘Who’ object listing nearby people and their data.
	Interaction Layer	
Input	(Intelligence layer outputs) (Device interaction inputs)	Interaction data from user and application interfaces
Output	Create Confirm Correct Delete Reset Relabel	Mapping interaction events to strict feedback actions, and outputting these actions through the API.

Table 6.1: *Abstract template for a software API for the layer model used in this dissertation.*

around the Five Ws context model and the user feedback actions seen in Chapter 5. The methods of each layer can be called independently of other layers. This modularity allows developers to use smaller components of the model for their own needs, rather than implementing the model as a whole.

Indeed, each layer could be treated as its own library, where developers might choose to use the sensing layer on its own (and obtain context data from physical, virtual and fused sensors through a sensing API), or feed their own data into the intelligence layer and use its outputs for their own applications. The flexibility of the layered approach means that developers need not be limited to using the entire architecture or nothing, but can pick and choose according to their application requirements.

6.6 The Five Ws as a Theory for Context?

Also in Chapter 2, we discussed how the Five Ws might form a good basis for a general context theory. A standard theory of context is lacking in the literature, and it would benefit the UbiComp community if a standard theory existed and its validity tested. We discussed how the intuitive notion of the Five Ws might be a good candidate for this.

How has the work in this dissertation furthered the theoretical understanding of context? Although the work is very empirical, there are certain elements of theory that have been advanced as a consequence of this work. Firstly, we have shown how the Five Ws can naturally describe context facets in multiple studies (see the layer model applied to the calendar in Chapter 3, and to place awareness in Chapter 4). Although we did prioritise some facets over others for our studies, there is huge potential for exploration of, for example, the 'Why' facet in relation to inferring human emotion and intent. Secondly, we have shown how context data sources and sensors can map to the Five Ws (Chapter 3) – both for physical and virtual sensors.

Of course there are elements of context that are difficult to define and model, and need further consideration in the development of a context theory. An example of this is concurrency – how do we define what might be considered parallel context, e.g. multiple instantiations of activity and continual transitions?

Nevertheless, there is a clear need for a standard context theory in UbiComp, and the work in this dissertation has shown how the Five Ws method might contribute towards such a theory.

6.7 Future Work

The work in this dissertation has contributed to the field of mobile context awareness, but there are still plenty of extensions to be made as well as directions for future work. To extend the context sensing work in Chapter 3, we could investigate further combinations of context sensors and the effects of their fusion on sensing performance. By considering software applications and services as virtual context sensors, there are innumerable possibilities for data fusion. We could further study the fidelity of unusual virtual sensors, e.g. measuring the performance of a user's Twitter feed or Facebook data as context sensors, and develop a set of standard metrics for measuring the quality of an entity as a context sensor. These metrics could then be used to aid designers in their choice of sensors in mobile context aware systems.

In Chapter 4, we evaluated our transition inference approach using the case study of place awareness. This is only one facet of context – the “Where” – and further work should be undertaken to extend detection inference from other facets, e.g. the “What” from activity transitions, which has been explored using body-worn motion sensors [92], but not mobile device sensors. The application of transition detection is broad, as it the transitions can be used as triggers for higher order processes. For example, we could study how responsive users are to unread message notifications when reminded on transition vs when the message arrives. We could even use the transitions as triggers in experience sampling; by prompting users for experience feedback on-transition rather than at random or periodically [51].

Also in Chapter 4, we developed a context inference and learning algorithm that: (i) was triggered by context transitions; and (ii) incorporated user assistance into the inference and learning process by prompting for user feedback when context reasoning was uncertain, i.e. a form of active machine learning. We applied our algorithm to place awareness once again, but we could further apply it to other use cases, e.g. museum tours, or incorporate other users' learned models of places to produce aggregate places – something that is beginning to gain interest in the literature, e.g. crowd sourced activity inference [132]. One of the limitations in our comparison of classification approaches in Chapter 4 was the duration of data collection, which goes some way to explaining the lack of difference between using an Hidden Markov Model and Bayesian Classifier. Thus, a further avenue of study lies in obtaining longer, fine-grained datasets upon which to test the efficacy of HMM-based classification.

In Chapter 5, we implemented our algorithm on the Android operating system and deployed in a field study with a view to observing how different approaches to and modes of user prompting affected feedback elicitation. Clearly other modes are possible for enabling feedback, e.g. gesture or more sophisticated tactile feedback, and further study could uncover the efficacy of these modes on obtaining intervention in the field. One of the key areas for future work here is the development of applications that can implicitly provide feedback from the user to the inference process, e.g. prompting the user within an application while another process is loading. It would be interesting to identify key moments such as this within applications and further measure how users respond to feedback requests during these moments.

There are two other areas for further study: (i) resource efficiency; and (ii) applications. For (i), we showed how our approach to place awareness is significantly more resource efficient than the state of the art, but further improvements could be made to reduce power and memory consumption. One direction of further study could be, for example, to develop sensing policies that optimally balance energy consumption with the requirement for sensed data; similar to approaches taken recently by Li *et al.* [136]. For (ii) – developing further mobile context aware applications – is an ongoing and popular research area. Much of the existing work in mobile context awareness concentrates on underlying technologies to enable context awareness, but comparatively little is done at the application level. We have presented a few prototype applications, but – with the exception of the "Where are you?" SMS application in Chapter 5 – none were systematically evaluated or user feedback sought. One of the primary areas for new research, therefore, is the further development of mobile context aware applications. Indeed, such developments would further take us towards the goal of making people's lives easier.

Bibliography

[1] Abowd, G., Atkeson, C., Hong, J., Long, S., and R. Cyberguide: A mobile context-aware tour guide. *Wireless Networks 3*, 5 (1997), 421–433.

[2] Abowd, G. D., and Mynatt, E. D. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction 7*, 1 (Mar. 2000), 29–58.

[3] Acid, S., de Campos, L. M., and Fernández, M. Evaluation methods and strategies for the interactive use of classifiers. *International Journal of Human-Computer Studies 70*, 5 (May 2012), 321–331.

[4] Adomavicius, G., and Tuzhilin, A. Context-aware recommender systems. *Proc. RecSys '08* (2008), 335.

[5] Adomavicius, G., Tuzhilin, A., Sankaranarayanan, R., and Sen, S. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems 23*, 1 (Jan. 2005), 103–145.

[6] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. Sentiment analysis of twitter data. In *Proc. of the Workshop on Languages in Social Media* (2011), Association for Computational Linguistics, pp. 30–38.

[7] Anagnostopoulos, T., Anagnostopoulos, C., Hadjiefthymiades, S., Kyriakakos, M., and Kalousis, A. Predicting the location of mobile users: a machine learning approach. In *Proc. ICPS '09* (London, United Kingdom, 2009), ACM, pp. 65–72.

[8] Anderson, I., and Muller, H. Context awareness via GSM signal strength fluctuation. In *In Adjunct Proc. Pervasive '06* (2006), vol. 01, pp. 27–31.

[9] Antifakos, S., Kern, N., Schiele, B., and Schwaninger, A. Towards improving trust in context-aware systems by displaying system confidence. In *Proc. MobileHCI '05* (2005), pp. 9–14.

[10] Antifakos, S., Schwaninger, A., and Schiele, B. Evaluating the effects of displaying uncertainty in context-aware applications. In *Proc. Ubicomp '04* (2004), pp. 54–69.

[11] Ashbrook, D., and Starner, T. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing 7*, 5 (Oct. 2003), 275–286.

[12] Asthana, A., Cravatts, M., and Krzyzanowski, P. An indoor wireless system for personalized shopping assistance. In *Workshop on Mobile Computing Systems and Applications* (1994), pp. 69–74.

[13] Azizyan, M., Constandache, I., and Choudhury, R. R. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proc. MobiCom '09* (2009), pp. 261–272.

[14] Bahl, P., and Padmanabhan, V. V. RADAR: An in-building RF-based user location and tracking system. In *Proc. INFOCOM '00* (2000), pp. 775–784.

[15] Baldauf, M., Dustdar, S., and Rosenberg, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing 2*, 4 (2007), 263–277.

[16] Bao, L., and Intille, S. Activity Recognition from User-annotated Acceleration Data. In *Proc. Pervasive '04* (2004), pp. 1–17.

[17] Barkhuus, L., Brown, B., Bell, M., Sherwood, S., Hall, M., and Chalmers, M. From awareness to repartee: sharing location within social groups. In *Proc. CHI '08* (2008), pp. 497–506.

[18] Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., and Seada, K. Fusing mobile, sensor, and social data to fully enable context-aware computing. In *Proc. HOTMOBILE '10* (2010), ACM Press, p. 60.

[19] Begole, J. B., Tang, J. C., and Hill, R. Rhythm Modeling, Visualizations, and Applications. *Proc. UIST '03 36*, 1 (2003), 28–46.

[20] Bellotti, V., and Edwards, K. Intelligibility and accountability: human considerations in context-aware systems. *Human Computer Interaction 16*, 2-4 (2001), 37–41.

[21] Benerecetti, M., Bouquet, P., and Ghidini, C. Contextual reasoning distilled. *Journal of Experimental & Theoretical Artificial Intelligence 12*, 3 (2000), 279–305.

[22] Benford, S., Seager, W., Flintham, M., Anastasi, R., Rowland, D., Humble, J., Fraser, D. S., Bowers, J., Tandavanitj, N., Adams, M., Farr, J. R., Oldroyd, A., and Sutton, J. The error of our ways: the experience of self-reported position in a location-based game. In *Proc. Ubicomp '04* (2004), pp. 70–87.

[23] Benisch, M., Sadeh, N., Kelley, P. G., and Cranor, L. F. Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs. *Personal and Ubiquitous Computing 15*, 7 (Dec. 2010), 679–694.

[24] Bennett, J., and Lanning, S. The Netflix Prize. In *Proc. KDDCup and Workshop '07* (2007), p. 35.

[25] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing 6*, 2 (Apr. 2010), 161–180.

[26] Bieber, G., Voskamp, J., and Urban, B. Activity Recognition for Everyday Life on Mobile Phones. In *Proc. UAHCI '09* (2009), pp. 289–296.

[27] Bishop, C. M. *Pattern recognition and machine learning.* Springer, 2006.

[28] Bleiholder, J., and Naumann, F. Data fusion. *ACM Computing Surveys 41*, 1 (Dec. 2008), 1–41.

[29] Bolliger, P. Redpin – adaptive, zero-configuration indoor localization through user collaboration. In *Proc. MELT '08* (2008), pp. 55–60.

[30] Bolliger, P., Partridge, K., Chu, M., and Langheinrich, M. Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling. In *LoCA*, T. Choudhury, A. J. Quigley, T. Strang, and K. Suginuma, Eds. Springer, 2009, pp. 37–51.

[31] Braden, R. RFC 1122: Requirements for Internet Hosts - Communication Layers. Tech. rep., 1989.

[32] Brezmes, T., Gorricho, J.-l., and Cotrina, J. Activity Recognition from Accelerometer Data on a Mobile Phone. In *Proc. IWANN '09* (2009), pp. 796–799.

[33] Brown, B., and Randell, R. Building a Context Sensitive Telephone: Some Hopes and Pitfalls for Context Sensitive Computing. *Proc. CSCW '04 13*, 3-4 (Aug. 2004), 329–345.

[34] Bunt, A., Lount, M., and Lauzon, C. Are Explanations Always Important? A Study of Deployed, Low-Cost Intelligent Interactive Systems. In *Proc. IUI '12* (2012), pp. 169–178.

[35] Chalmers, D. Classification and Use of Context. In *Sensing and Systems in Pervasive Computing*, Undergraduate Topics in Computer Science. Springer London, London, 2011, pp. 67–83.

[36] Chalmers, D., Dulay, N., and Sloman, M. A framework for contextual mediation in mobile and ubiquitous computing applied to the context-aware adaptation of maps. *Personal and Ubiquitous Computing 8*, 1 (Feb. 2004), 1–18.

[37] Chalmers, D., Dulay, N., and Sloman, M. Meta data to support context aware mobile applications. In *Proc. MDM '04* (2004), pp. 199–210.

[38] Chen, G., and Kotz, D. A Survey of Context-Aware Mobile Computing Research. Tech. rep., Dartmouth College, 2000.

[39] Chen, H., Finin, T., and Joshi, A. An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review 18*, 3 (Sept. 2003), 197–207.

[40] Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. Developing a Context-aware Electronic Tourist Guide : Some Issues and Experiences. In *Proc. CHI '00* (2000), pp. 17–24.

[41] Chon, Y., Lane, N., Li, F., Cha, H., and Zhao, F. Automatically Characterizing Places with Opportunistic CrowdSensing using Smartphones. In *Proc. Ubicomp '12* (2012), pp. 481–490.

[42] Chon, Y., Talipov, E., and Cha, H. Autonomous Management of Everyday Places for a Personalized Location Provider. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,. 42*, 99 (2011), 1–14.

[43] Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., and Others. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing 7*, 2 (2008), 32–41.

[44] Choujaa, D., and Dulay, N. TRAcME: Temporal Activity Recognition Using Mobile Phone Data. In *Proc. EUC '08* (Dec. 2008), pp. 119–126.

[45] Choujaa, D., and Dulay, N. Using screenplays as a source of context data. In *Proc. SRMC '08* (2008), ACM Press, pp. 13–19.

[46] Choujaa, D., and Dulay, N. Activity inference through sequence alignment. In *Proc. LoCA '09* (2009), pp. 19–36.

[47] Choujaa, D., and Dulay, N. Predicting human behaviour from selected mobile phone data points. In *Proc. Ubicomp '10* (2010), pp. 105–108.

[48] Chung, J., Donahoe, M., Schmandt, C., Kim, I.-J., Razavai, P., and Wiseman, M. Indoor location sensing using geo-magnetism. In *Proc. MobiSys '11* (New York, New York, USA, 2011), ACM Press, pp. 141–154.

[49] Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine Learning 15*, 2 (May 1994), 201–221.

[50] Consolvo, S., Mcdonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., Klasnja, P., Lamarca, A., Legrand, L., Libby, R., Smith, I., and Landay, J. A. Activity Sensing in the Wild: A Field Trial of UbiFit Garden. In *Proc. CHI '08* (2008), pp. 1797–1806.

[51] Consolvo, S., and Walker, M. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing 2*, 2 (Apr. 2003), 24–31.

[52] Corral-Verdugo, V. Dual Realities of Conservation Behavior: Self-Reports vs Observations of Re-Use and Recycling Behaviour. *Journal of Environmental Psychology 17* (1997), 135–145.

[53] Costanza, E., Ramchurn, S. D., and Jennings, N. R. Understanding Domestic Energy Consumption through Interactive Visualisation : a Field Study. In *Proc. Ubicomp '12* (2012), pp. 216–225.

[54] Daamen, W., and Hoogendoorn, S. Free Speed Distributions for Pedestrian Traffic. In *Proc TRB '06* (2006).

[55] Dey, A., Abowd, G., and Salber, D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction 16*, 2 (Dec. 2001), 97–166.

[56] Dey, A., Wac, K., and Ferreira, D. Getting Closer: An Empirical Investigation of the Proximity of Users to their Smart Phones. In *Proc. Ubicomp '11* (2011), pp. 163–172.

[57] Dey, A. K. Understanding and Using Context. *Personal and Ubiquitous Computing 5*, 1 (2001), 4–7.

[58] Dey, A. K., and Abowd, G. D. CybreMinder : A Context-Aware System for Supporting Reminders. In *Proc. HUC '00* (2000), pp. 172–186.

[59] Dey, A. K., and Abowd, G. D. Towards a better understanding of context and context-awareness. In *CHI'2000 Workshop on the What, Who, Where, When, and How of Context-Awareness* (2000).

[60] Dey, A. K., and Mankoff, J. Designing mediation for context-aware applications. *ACM Transactions on Computer-Human Interaction 12*, 1 (Mar. 2005), 53–80.

[61] Dix, A., Rodden, T., Davies, N., Trevor, J., Friday, A., and Palfreyman, K. Exploiting space and location as a design framework for interactive mobile systems. *ACM Transactions on Computer-Human Interaction 7*, 3 (Sept. 2000), 285–321.

[62] Dourish, P. What we talk about when we talk about context. *Personal and Ubiquitous Computing 8*, 1 (2004), 19–30.

[63] Dourish, P., Adler, A., and Smith, B. Organising user interfaces around reflective accounts. In *Reflection'96* (1996), pp. 235–244.

[64] Duda, R. Sequential k-Means Clustering, 2008.

[65] Eagle, N., and Pentland, A. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing 4*, 2 (2005), 28–34.

[66] Eagle, N., and Pentland, A. S. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology 63*, 7 (2009), 1057–1066.

[67] Eagle, N., and (Sandy) Pentland, A. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing 10*, 4 (Nov. 2005), 255–268.

[68] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., and Estrin, D. Diversity in smartphone usage. In *Proc. MobiSys '10* (New York, New York, USA, 2010), ACM Press, p. 179.

[69] Figo, D., Diniz, P. C., Ferreira, D. R., and Cardoso, J. M. P. Preprocessing Techniques for Context Recognition from Accelerometer Data. *Personal and Ubiquitous Computing 14*, 7 (Mar. 2010), 645–662.

[70] Fisher, R., and Simmons, R. Smartphone Interruptibility Using Density-Weighted Uncertainty Sampling with Reinforcement Learning. In *Proc. ICMLA '11* (Dec. 2011), pp. 436–441.

[71] Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., and Yang, J. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction 12*, 1 (Mar. 2005), 119–146.

[72] Fogarty, J., Lai, J., Christensen, J., and Christensen, H. Presence versus Availability: the Design and Evaluation of a Context-aware Communication Client. *International Journal of Human-Computer Studies 61*, 3 (Sept. 2004), 299–317.

[73] Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., and Landay, J. A. MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones. In *Proc. MobiSys '07* (2007), pp. 57–70.

[74] Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., and Landay, J. UbiGreen: Investigating a Mobile Tool for Tracking and Supporting Green Transportation Habits. In *Proc. CHI '09* (2009), pp. 1043–1052.

[75] Garzonis, S. *Mobile Service Awareness via Auditory Notifications*. PhD thesis, University of Bath, 2010.

[76] Gaver, B. Provocative awareness. *Proc. CSCW '02 11*, 3 (2002), 475–493.

[77] Gellersen, H., Lukowicz, P., Beigl, M., and Riedel, T. Cooperative Relative Positioning. *IEEE Pervasive Computing 9*, 4 (2010), 78–89.

[78] Gellersen, H. W., Schmidt, A., and Beigl, M. Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts. *Mobile Networks and Applications 7*, 5 (2002), 341–351.

[79] Geyer, W., Richter, H., and Abowd, G. Towards a Smarter Meeting Record—Capture and Access of Meetings Revisited. *Multimedia Tools and Applications 27*, 3 (2005), 393–410.

[80] Greenberg, S. Context as a dynamic construct. *Human-Computer Interaction 16*, 2-4 (2001), 257–268.

[81] Griswold, W., Shanahan, P., and Brown, S. ActiveCampus: experiments in community-oriented ubiquitous computing. *Computer 37*, 10 (2004), 73–81.

[82] Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. CHI '01* (2001), pp. 458–465.

[83] Gu, T., Pung, H. K., and Zhang, D. Q. A Service-Oriented Middleware for Building Context-Aware Services. *Journal of Network and Computer Applications 28*, 1 (2005), 1–18.

[84] Gütl, C., and García-Barrios, V. Semantic Meeting Information Application: A Contribution for Enhanced Knowledge Transfer and Learning in Companies. In *Proc. ICL '05* (Villach, Austria, 2005).

[85] Hall, D. L., Member, S., and Llinas, J. An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE 85*, 1 (1997), 6–23.

[86] Harrison, S., and Dourish, P. Re-Place-ing Space : The Roles of Place and Space in Collaborative Systems. In *Proc. CSCW '96* (1996), pp. 64–76.

[87] Henricksen, K., Indulska, J., and Rakotonirainy, A. Modeling Context Information in Pervasive Computing Systems. In *Proc. Pervasive '02* (2002), pp. 167–180.

[88] Hicks, J., Ramanathan, N., Kim, D., Monibi, M., Selsky, J., Hansen, M., and Estrin, D. AndWellness: An Open Mobile System for Activity and Experience Sampling. In *Proc. WH '10* (2010), pp. 34–43.

[89] Hightower, J. From position to place. In *Workshop on Location Aware Computing at Ubicomp '03* (2003), pp. 10–13.

[90] Hightower, J., Consolvo, S., LaMarca, A., Smith, I., and Hughes, J. Learning and recognizing the places we go. In *Proc. UbiComp '05* (2005), pp. 903–903.

[91] Hinckley, K., and Horvitz, E. Toward more sensitive mobile phones. In *Proc. UIST '01* (2001), vol. 3, ACM Press, pp. 191–192.

[92] Ho, J., and Intille, S. S. Using Context-aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proc. CHI '05* (2005), pp. 909–918.

[93] Hofer, T., and Schwinger, W. Context-awareness on mobile devices-the hydrogen approach. In *Proc. HICSS '03* (2003), vol. 43.

[94] Hong, J.-Y., Suh, E.-H., and Kim, S.-J. Context-aware systems: A literature review and classification. *Expert Systems with Applications 36*, 4 (2009), 8509–8522.

[95] Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., and Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing 5*, 3 (June 2009), 236–252.

[96] Hong, Y.-J., Kim, I.-J., Ahn, S. C., and Kim, H.-G. Mobile health monitoring system based on activity recognition using accelerometer. *Simulation Modelling Practice and Theory 18*, 4 (Apr. 2010), 446–455.

[97] Horvitz, E., Koch, P., Kadie, C. C., and Jacobs, A. Coordinate: Probabilistic forecasting of presence and availability. In *Proc. UAI '02* (2002), pp. 224–233.

[98] Horvitz, E., Koch, P., Sarin, R., Apacible, J., and Subramani, M. Bayesphone: Precomputation of context-sensitive policies for inquiry and action in mobile devices. In *Proc. UM '05* (2005), pp. 251 – 260.

[99] Hsu, J.-m., Wu, W.-j., Chang, I.-r., and Chung, N. Ubiquitous Multimedia Information Delivering Service for Smart Home. In *Proc. MUE '07* (2007), pp. 341–346.

[100] Hudson, J. M., Christensen, J., Kellogg, W. A., and Erickson, T. " I'd Be Overwhelmed, But It's Just One More Thing to Do " Availability and Interruption in Research Management. In *Proc. CHI '02* (2002), no. 4, pp. 97–104.

[101] Hudson, S., Fogarty, J., Atkeson, C., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J., and Yang, J. Predicting human interruptibility with sensors: a Wizard of Oz feasibility study. In *Proc. CHI '03* (Ft. Lauderdale, Florida, USA, 2003), ACM, pp. 257–264.

[102] Huebscher, M., McCann, J., and Dulay, N. Fusing Multiple Sources of Context Data of the Same Context Type. In *Proc. ICHIT '06* (Nov. 2006), pp. 406–415.

[103] Indulska, J., and Sutton, P. Location management in pervasive systems. In *Proc. ACSW Frontiers '03* (Adelaide, Australia, 2003), pp. 143 –151.

[104] Jacucci, G., Oulasvirta, A., Ilmonen, T., Evans, J., and Salovaara, A. Comedia: mobile group media for active spectatorship. In *Proc. CHI '07* (2007), pp. 1273–1282.

[105] Jameson, A., and Riedl, J. Introduction to the Transactions on Interactive Intelligent Systems. *ACM Transactions on Interactive Intelligent Systems 1*, 1 (Oct. 2011), 1–6.

[106] Jiang, Y., Pan, X., Li, K., Lv, Q., Dick, R. P., Hannigan, M., and Shang, L. ARIEL: Automatic Wi-Fi based Room Fingerprinting for Indoor Localization. In *Proc. Ubicomp '12* (2012), pp. 441–450.

[107] Jin, X., Li, Y., Mah, T., and Tong, J. Sensitive webpage classification for content advertising. In *Proc. 1st international workshop on Data mining and audience intelligence for advertising* (2007), ACM, pp. 28–33.

[108] Jung, J., Lee, H., and Choi, K. S. Contextualized Recommendation Based on Reality Mining From Mobile Subscribers. *Cybernetics and Systems 40*, 2 (Feb. 2009), 160–175.

[109] Kaenampornpan, M., and O'Neill, E. Modelling context: An activity theory approach. In *Proc. USAI '04* (2004), pp. 367–374.

[110] Kang, J. H., Welbourne, W., Stewart, B., and Borriello, G. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review 9*, 3 (July 2005), 58.

[111] Kang, S., Lee, J., Jang, H., Lee, H., Lee, Y., Park, S., Park, T., and Song, J. Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *MobiSys '08* (2008), pp. 267–280.

[112] Kapoor, A., and Horvitz, E. Experience Sampling for Building Predictive User Models: a Comparative Study. In *Proc. CHI '08* (2008), pp. 657–666.

[113] Karvonen, K. The beauty of simplicity. In *Proc. CUU '00* (2000), pp. 85–90.

[114] Kern, N., and Schiele, B. Towards personalized mobile interruptibility estimation. In *Proc. LoCA '06* (2006), pp. 134–150.

[115] Kim, D. H., Han, K., and Estrin, D. Employing User Feedback for Semantic Location Services. In *Proc. Ubicomp '11* (2011), pp. 217–226.

[116] Kim, D. H., Hightower, J., Govindan, R., and Estrin, D. Discovering Semantically Meaningful Places from Pervasive RF-Beacons. In *Proc. UbiComp '09* (2009), pp. 21–30.

[117] Kim, D. H., Kim, Y., Estrin, D., and Srivastava, M. B. SensLoc : Sensing Everyday Places and Paths using Less Energy. In *Proc. SenSys '10* (2010), pp. 43–56.

[118] Kjaergaard, M. B. Location-Based services on Mobile Phones: Minimizing Power Consumption. *IEEE Pervasive Computing 11*, 1 (2012), 67–73.

[119] Kjaergaard, M. B., Langdal, J., Godsk, T., and Toftkjaer, T. EnTracked : Energy-Efficient Robust Position Tracking for Mobile Devices. In *Proc. MobiSys '09* (2009), pp. 221–234.

[120] Kjaergaard, M. B., Wirz, M., Roggen, D., and Troster, G. Detecting Pedestrian Flocks by Fusion of Multi-Modal Sensors in Mobile Phones. In *Proc. Ubicomp '12* (2012), pp. 240–249.

[121] Kohonen, T. The Self-Organizing Map. *Proc. of the IEEE 78*, 9 (1990), 1464–1480.

[122] Korpipää, P., and Mäntyjärvi, J. An ontology for mobile device sensor-based context awareness. In *Proc. CONTEXT '03* (2003), pp. 451–458.

[123] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., and Malm, E.-J. Managing context information in mobile devices. *Pervasive Computing, IEEE 2*, 3 (July 2003), 42–51.

[124] Kostakos, V., and O'Neill, E. Cityware: Urban computing to bridge online and real-world social networks. In *Handbook of research on urban informatics: the practice and promise of the real-time city*, M. Foth, Ed. Hershey,Philadelphia, USA, 2008.

[125] Krause, A., Smailagic, A., and Siewiorek, D. Context-aware mobile computing: learning context- dependent personal preferences from a wearable sensor array. *IEEE Transactions on Mobile Computing 5*, 2 (Feb. 2006), 113–127.

[126] Krumm, J., and Horvitz, E. Locadio: Inferring motion and location from Wi-Fi signal strengths. In *Proc. Mobiquitous '04* (2004), vol. 2004, pp. 4–13.

[127] Kuznetsov, S., Odom, W., James, P., and Paulos, E. Nurturing Natural Sensors. In *Proc. Ubicomp '11* (2011), pp. 227–236.

[128] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity Recognition using Cell Phone Accelerometers. *SIGKDD Explorer Newsletter 12*, 2 (2010), 74–82.

[129] Laasonen, K., Raento, M., and Toivonen, H. Adaptive On-Device Location Recognition. In *Proc. Pervasive '04* (2004), pp. 287–304.

[130] LaMarca, A., Chawathe, Y., and Consolvo, S. Place lab: Device positioning using radio beacons in the wild. In *Proc. Pervasive '05* (2005), pp. 116–133.

[131] Lane, N., Miluzzo, E., and Lu, H. A survey of mobile phone sensing. *IEEE Communications Magazine 48*, 9 (2010), 140–150.

[132] Lane, N. D., Xu, Y., Lu, H., Hu, S., Choudhury, T., Campbell, A. T., and Zhao, F. Enabling Large-scale Human Activity Inference on Smartphones using Community Similarity Networks (CSN). In *Proc. Ubicomp '11* (2011), pp. 355–364.

[133] Lee, C., Yoon, G., and Han, D. A Probabilistic Place Extraction Algorithm Based on a Superstate Model. *IEEE Transactions on Mobile Computing* (2012), 1–13.

[134] Lewis, D., and Gale, W. A sequential algorithm for training text classifiers. In *Proc. SIGIR '94* (1994), pp. 3–12.

[135] Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., Zhao, F., and Beijing, C. A Reliable and Accurate Indoor Localization Method Using Phone Inertial

Sensors. In *Proc. Ubicomp '12* (New York, New York, USA, 2012), ACM Press, pp. 421–430.

[136] Li, X., Cao, H., Chen, E., and Tian, J. Learning to Infer the Status of Heavy-Duty Sensors for Energy-Efficient Context-Sensing. *ACM Transactions on Intelligent Systems and Technology 3*, 2 (2012), 35:1–35:23.

[137] Liao, L., Fox, D., and Kautz, H. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research 26*, 1 (Jan. 2007), 119–134.

[138] Lieberman, H., and Selker, T. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal 39*, 3.4 (2000), 617–632.

[139] Lim, B., and Dey, A. Toolkit to support intelligibility in context-aware applications. In *Ubicomp '10* (2010), ACM, pp. 13–22.

[140] Lim, B., and Dey, A. Investigating intelligibility for uncertain context aware applications. In *Proc. Ubicomp '11* (2011), pp. 415–424.

[141] Lim, B. Y., and Dey, A. K. Assessing demand for intelligibility in context-aware applications. In *Proc. Ubicomp '09* (New York, New York, USA, 2009), ACM Press, pp. 195–204.

[142] Lim, B. Y., and Dey, A. K. Design of an Intelligible Mobile Context-Aware Application. In *Proc. MobileHCI '11* (2011), pp. 157–166.

[143] Lim, B. Y., Dey, A. K., and Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proc. CHI '09* (New York, New York, USA, 2009), ACM Press, pp. 2119–2128.

[144] Lovett, T., and O'Neill, E. Capturing Transitions Between Users' Semantically Meaningful Places Using Mobile Devices. In *In Proc. MCSS at MobiSys '12* (2012), pp. 11–16.

[145] Lovett, T., O'Neill, E., Irwin, J., and Pollington, D. The Calendar as a Sensor: Analysis and Improvement using Data Fusion with Social Networks and Location. In *Proc. Ubicomp '10* (2010), pp. 3–12.

[146] Lu, H., Brush, A. B., Priyantha, B., Karlson, A. K., and Liu, J. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *Proc. Pervasive '11* (2011), pp. 188–205.

[147] Lukkari, J., Korhonen, J., and Ojala, T. SmartRestaurant – Mobile Payments in Context-Aware. In *Proc. ICEC '04* (2004), pp. 575–582.

[148] Luštrek, M., and Kaluža, B. Fall Detection and Activity Recognition with Machine Learning. *Informatica 33* (2009), 205–212.

[149] MacKay, D. J. C. Information-Based Objective Functions for Active Data Selection. *Neural Computation 4*, 4 (July 1992), 590–604.

[150] Mayrhofer, R. An architecture for context prediction. In *Advances in Pervasive Computing*, A. Ferscha, H. Hoertner, and G. Kotsis, Eds. Austrian Computer Society (OCG), 2005, pp. 65–72.

[151] Mayrhofer, R., Radi, H., and Ferscha, A. Recognizing and Predicting Context by Learning from User Behavior. In *Proc. MoMM '03* (2003), vol. 1, pp. 25–35.

[152] McCrickard, D., and Chewar, C. Attuning notification design to user goals and attention costs. *Communications of the ACM 46*, 3 (2003), 67–72.

[153] Miluzzo, E., Cornelius, C., Ashwin, R., Choudhury, T., Liu, Z., and Campbell, A. T. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proc. MobiSys '10* (2010), pp. 5–20.

[154] Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., and Campbell, A. T. Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In *Proc. SenSys '08* (2008), pp. 337–350.

[155] Montoliu, R., Blom, J., and Gatica-Perez, D. Discovering places of interest in everyday life from smartphone data. *Multimedia Tools and Applications* (Jan. 2012), 1–29.

[156] Montoliu, R., and Gatica-Perez, D. Discovering Human Places of Interest from Multimodal Mobile Phone Data. In *Proc. MUM '10* (2010), pp. 1–10.

[157] Moran, T., and Dourish, P. Introduction to this special issue on context-aware computing. *Human Computer Interaction 16*, 2-4 (2001), 87–95.

[158] Muñoz, M. A., Gonzalez, V. M., Rodríguez, M., and Favela, J. Supporting Context-Aware Collaboration in a Hospital : An Ethnographic Informed Design. In *Groupware: Design, Implementation and Use*, J. Favela and D. Decouchant, Eds. Springer Berlin / Heidelberg, 2003, pp. 330–344.

[159] Muir, B. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics 37*, 11 (1994), 1905–1922.

[160] Mynatt, E., and Tullio, J. Inferring calendar event attendance. In *Proc. IUI '01* (Santa Fe, New Mexico, United States, 2001), ACM, pp. 121–128.

[161] Mynatt, E. D., Back, M., Want, R., Baer, M., and Ellis, J. B. Designing audio aura. In *Proc. CHI '98* (1998), pp. 566–573.

[162] Neustaedter, C., Brush, A. J. B., and Greenberg, S. The calendar is crucial: Coordination and awareness through the family calendar. *ACM Transactions on Computer-Human Interaction 16*, 1 (2009), 1–48.

[163] Ni, L. M., Liu, Y., Lau, Y. C., and Patil, A. P. LANDMARC: Indoor Location Sensing Using Active RFID. *Wireless Networks 10*, 6 (Nov. 2004), 701–710.

[164] Nurmi, P., and Koolwaaij, J. Identifying meaningful locations. In *Proc. Mobile and Ubiquitous Systems: Networking & Services* (2006), pp. 1 – 8.

[165] O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. ICWSM '10* (2010), pp. 122–129.

[166] Oh, Y., Yoon, H., and Woo, W. Simulating Context-Aware Systems based on Personal Devices. In *Proc. ISUVR '06* (2006), pp. 49–52.

[167] Oliver, E., and Keshav, S. An empirical approach to smartphone energy level prediction. In *Proc. Ubicomp '11* (2011), pp. 345–354.

[168] O'Neill, E., Kostakos, V., Kindberg, T., Fatah, A., Penn, A., Fraser, D. S., and Jones, T. Instrumenting the City : Developing Methods for Observing and Understanding the Digital Cityscape. In *Proc. Ubicomp '06* (2006), pp. 315 – 332.

[169] Padovitz, A., Loke, S., Zaslavsky, A., Burg, B., and Bartolini, C. An approach to data fusion for context awareness. In *Modelling and Using Context*, A. Dey, B. Kokinov, D. Leake, and R. Turner, Eds. Springer Berlin / Heidelberg, 2005, pp. 9–63.

[170] Palen, L., and Grudin, J. Discretionary adoption of group support software: lessons from calendar applications. In *Implementing collaboration technologies in industry: case examples and lessons learned.* Spring-Verlag, London, UK, 2003, pp. 159–179.

[171] Pang, B., and Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. 42nd annual meeting*

on Association for Computational Linguistics (2004), Association for Computational Linguistics, p. 271.

[172] Pang, B., and Lee, L. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval 2*, 1-2 (2008), 1–135.

[173] Park, J.-g., Charrow, B., Curtis, D., Battat, J., Minkov, E., Hicks, J., Teller, S., and Ledlie, J. Growing an organic indoor location system. In *Proc. MobiSys '10* (2010), pp. 271–284.

[174] Pascoe, J., Ryan, N., and Morse, D. Issues in Developing Context-Aware Computing. In *Proc. HUC '99* (1999), pp. 208–221.

[175] Patel, S., Kientz, J., Hayes, G., Bhat, S., and Abowd, G. Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *Proc. UbiComp '06*, P. Dourish and A. Friday, Eds. Springer Berlin / Heidelberg, 2006, pp. 123 – 140.

[176] Patterson, D., Liao, L., Fox, D., and Kautz, H. Inferring high-level behavior from low-level sensors. In *Proc. Ubicomp '03*, A. Dey, A. Schmidt, and J. McCarthy, Eds. Springer Berlin / Heidelberg, 2003, pp. 73–89.

[177] Paulos, E., and Goodman, E. The Familiar Stranger: Anxiety, Comfort, and Play in Public Places. In *Proc. CHI '04* (2004), pp. 223–230.

[178] Payne, S. J. Understanding Calendar Use. *Human-Computer Interaction 8*, 2 (1993), 83–100.

[179] Peddinti, S., and Saxena, N. On the limitations of query obfuscation techniques for location privacy. In *Proc. Ubicomp '11* (2011), pp. 187–196.

[180] Pedersen, E., and Sokoler, T. AROMA: abstract representation of presence supporting mutual awareness. In *Proc. CHI '97* (1997), pp. 51–58.

[181] Pirkl, G., and Lukowicz, P. Robust, Low Cost Indoor Positioning Using Magnetic Resonant Coupling. In *Proc. Ubicomp '12* (2012), pp. 431–440.

[182] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. The Cricket location-support system. In *Proc. MobiCom '00* (2000), pp. 32–43.

[183] Ra, M.-R., Priyantha, B., Kansal, A., and Liu, J. Improving energy efficiency of personal sensing applications with heterogeneous multi-processors. In *Proc. Ubicomp '12* (New York, New York, USA, 2012), ACM Press, pp. 1–10.

[184] Rabbi, M., Ali, S., Choudhury, T., and Berke, E. Passive and In-Situ assessment of mental and physical well-being using mobile sensors. In *Proc. Ubicomp '11* (2011), pp. 385–394.

[185] Rabiner, L. L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[186] Rachuri, K. K., Mascolo, C., and Musolesi, M. Energy-Accuracy Tradeoffs in Querying Sensor Data for Continuous Sensing Mobile Systems. In *Workshop on Mobile Context-Awareness, Ubicomp '10* (2010).

[187] Rachuri, K. K., Musolesi, M., and Mascolo, C. EmotionSense: a mobile phones based adaptive platform for experimental social psychology research. In *Proc. Ubicomp '10* (2010), pp. 281–290.

[188] Raento, M., Oulasvirta, a., Petit, R., and Toivonen, H. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing 4*, 2 (Apr. 2005), 51–59.

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

[190] Ravi, N., Scott, J., Han, L., and Iftode, L. Context-aware Battery Management for Mobile Phones. In *Proc. PerCom '08* (Mar. 2008), pp. 224–233.

[191] Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., and Srivastava, M. Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensor Networks 6*, 2 (Feb. 2010), 1–27.

[192] Rekimoto, J., Miyaki, T., and Ishizawa, T. LifeTag: WiFi-based Continuous Location Logging for Life Pattern Analysis. In *Proc. LoCA '07* (2007), pp. 35–49.

[193] Riboni, D., and Bettini, C. COSAR: Hybrid Reasoning for Context-aware Activity Recognition. *Personal and Ubiquitous Computing 15*, 3 (Aug. 2011), 271–289.

[194] Robertson, D. W. A Note on the Classical Origin of "Circumstances". *Studies in Philology 43*, 1 (1946), 6–14.

[195] Rogers, Y., Connelly, K., Tedesco, L., Hazlewood, W., Kurtz, A., Hall, R. E., Hursey, J., and Toscos, T. Why It 's Worth the Hassle: The Value of In-Situ Studies When Designing Ubicomp. In *Proc. UbiComp '07* (2007), pp. 336–353.

[196] Rosenthal, S., Dey, A., and Veloso, M. Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models. In *Proc. Pervasive '11* (2011), pp. 170–187.

[197] Rosenthal, S., Veloso, M., and Dey, A. K. Acquiring Accurate Human Responses to Robots' Questions. *International Journal of Social Robotics* (Feb. 2012), 1–13.

[198] Rosenthal, S. L., and Dey, A. K. Towards Maximizing the Accuracy of Human-Labeled Sensor Data. In *Proc. IUI '10* (2010), pp. 259–268.

[199] Russell, S., and Norvig, P. *Artificial intelligence: a modern approach*, 3rd ed. Prentice Hall Press, 2009.

[200] Sadilek, A., and Kautz, H. Modeling Success , Failure , and Intent of Multi-Agent Activities Under Severe Noise. In *Mobile Context Awareness*, T. Lovett and E. O'Neill, Eds. Springer, 2012, pp. 9–36.

[201] Sawhney, N., and Schmandt, C. Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments. *ACM Transactions on Computer-Human Interaction 7*, 3 (Sept. 2000), 353–383.

[202] Schilit, B., Adams, N., and Want, R. Context-aware computing applications. In *First Workshop on Mobile Computing Systems and Applications, 1994. WMCSA 1994.* (1994), vol. pp, IEEE Comput. Soc. Press, pp. 85–90.

[203] Schilit, B., Hilbert, D., and Trevor, J. Context-aware communication. *IEEE Wireless Communications 9*, 5 (2002), 46–54.

[204] Schilit, B., and Theimer, M. Disseminating Active Map Information to Mobile Hosts. *IEEE Network 8*, 5 (1994), 22–32.

[205] Schmidt, A. Implicit human computer interaction through context. *Personal and Ubiquitous Computing 4*, 2-3 (2000), 191–199.

[206] Schmidt, A. Interactive context-aware systems interacting with ambient intelligence. In *Ambient intelligence.* 2005, pp. 159–178.

[207] Schmidt, A., Asante, K., and Takaluoma, A. Advanced Interaction in Context. In *Handheld and Ubiquitous Computing*, H.-W. Gellersen, Ed. Springer Berlin / Heidelberg, 1999, pp. 89–101.

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

[209] Schmidt, K. The problem with 'awareness': Introductory remarks on 'awareness in CSCW'. *Proc. CSCW '02 11*, 4 (2002), 285–298.

[210] Settles, B. Active Learning Literature Survey. Tech. Rep. 2, University of Wisconsin-Madison, 2009.

[211] Shiels, M. Mobile application sales to reach '$17.5bn by 2012' (http://news.bbc.co.uk/1/hi/technology/8571210.stm [Accessed November 2012]), 2010.

[212] Siewiorek, D., and Smailagic, A. Sensay: A context-aware mobile phone. In *ISWC '03* (2003), p. 248.

[213] Sigg, S., Haseloff, S., and David, K. An Alignment Approach for Context Prediction Tasks in UbiComp environments. *IEEE Pervasive Computing 9*, 4 (2010), 90–97.

[214] Sohn, T., Li, K. A., Lee, G., Smith, I., Scott, J., and Griswold, W. G. Place-Its : A Study of Location-Based Reminders on Mobile Phones. In *Proc. Ubicomp '05* (2005), pp. 232–250.

[215] Sohn, T., Varshavsky, A., LaMarca, A., Chen, M. Y., Choudhury, T., Smith, I., Consolvo, S., Hightower, J., Griswold, W. G., and de Lara, E. Mobility detection using everyday GSM traces. In *Proc. Ubicomp '06* (2006), pp. 212–224.

[216] Spertus, E. Smokey: Automatic recognition of hostile messages. In *Proc. of the National Conference on Artificial Intelligence* (1997), JOHN WILEY & SONS LTD, pp. 1058–1065.

[217] Stikic, M., Larlus, D., Ebert, S., Member, S., and Schiele, B. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *Analysis 33*, 12 (2011), 2521–2537.

[218] Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. Toward harnessing user feedback for machine learning. In *Proc. IUI '07* (New York, New York, USA, 2007), ACM Press, pp. 82 – 91.

[219] Stumpf, S., Rajaram, V., Li, L., Wong, W.-k., Burnett, M., Dietterich, T., Sullivan, E., and Herlocker, J. Interacting Meaningfully with Machine Learning Systems: Three Experiments. *International Journal of Human-Computer Studies 67*, 8 (2009), 639–662.

[220] Stumpf, S., Sullivan, E., Fitzhenry, E., Oberst, I., Wong, W.-k., and Burnett, M. Integrating Rich User Feedback into Intelligent User Interfaces. In *Proc. IUI '08* (2008), pp. 50–59.

[221] Sumi, Y., Etani, T., Fels, S., Simonet, N., Kobayashi, K., and Mase, K. C-MAP: Building a Context-Aware Mobile Assistant for Exhibition Tours. In *Community Computing and Support Systems, Social Interaction in Networked Communities*. 1998, pp. 137–154.

[222] Tang, J. C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., and Bhalodia, J. ConNexus to awarenex. In *Proc. CHI '01* (2001), no. Im, pp. 221–228.

[223] Tapia, E. M., Intille, S. S., and Larson, K. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive Computing*, A. Ferscha and F. Mattern, Eds. 2004, pp. 158–175.

[224] Tatemura, J. Virtual reviewers for collaborative exploration of movie reviews. In *Proc. IUI '00* (2000), ACM, pp. 272–275.

[225] Trewin, S., and Zimmermann, G. Abstract User Interface Representations : How Well do they Support Universal Access ? In *Proc. CUU '03* (2003), pp. 77–84.

[226] Tullio, J. *Exploring the design and use of forecasting groupware applications with an augmented shared calendar.* Ph.D. thesis, 2005.

[227] Tullio, J., Goecks, J., Mynatt, E. D., and Nguyen, D. H. Augmenting shared personal calendars. In *Proc. UIST '02* (New York, New York, USA, 2002), vol. 4, ACM, pp. 11–20.

[228] Tullio, J., and Mynatt, E. Use and implications of a shared, forecasting calendar. In *Proc. INTERACT '07* (Rio de Janeiro, Brazil, 2007), Springer-Verlag, pp. 269–282.

[229] van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B., Kasteren, T. V., and Kr, B. Accurate activity recognition in a home setting. In *Proc. Ubicomp '08* (New York, New York, USA, 2008), ACM Press, pp. 1–9.

[230] van Laerhoven, K. Combining the Self-Organizing Map and K-Means Clustering for On-line Classification of Sensor Data. In *Proc. ICANN '01* (2001), pp. 464–469.

[231] Van Laerhoven, K., and Cakmakci, O. What shall we teach our pants? In *Proc. ISWC '00* (2000), no. c, pp. 77–83.

[232] Wang, J., Chen, G., and Kotz, D. A Sensor-fusion Approach for Meeting Detection. In *Proc. of the Workshop on Context Awareness at MobiSys '04* (Boston, MA, USA, 2004).

[233] Wang, Y., Lin, J., Annavaram, M., Jacobson, Q. a., Hong, J., Krishnamachari, B., and Sadeh, N. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proc. MobiSys '09* (New York, New York, USA, 2009), ACM Press, pp. 179–192.

[234] Want, R., Hopper, A., Falcão, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems 10*, 1 (Jan. 1992), 91–102.

[235] Ward, A., Jones, A., and Hopper, A. A new location technique for the active office. *IEEE Personal Communications 4*, 5 (1997), 42–47.

[236] Weiser, M. The computer for the 21st century. *Scientific American* (1991), 94 – 104.

[237] Weiser, M., and Brown, J. S. Designing Calm Technology. Tech. rep., 1995.

[238] Westermann, U., and Jain, R. Toward a Common Event Model for Multimedia Applications. *IEEE MultiMedia 14*, 1 (2007), 19–29.

[239] Woo, W., Korea, S., and Oh, Y. A Unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness. In *Proc. UCS '04* (2005), pp. 192–202.

[240] Woodman, O., and Harle, R. Pedestrian localisation for indoor environments. In *Proc. Ubicomp '08* (New York, New York, USA, 2008), ACM Press, pp. 114 – 123.

[241] Wright, M., Lin, C., O'Neill, E., Cosker, D., and Johnson, P. 3D gesture recognition: an evaluation of user and system performance. In *Pervasive Computing* (2011), pp. 249–313.

[242] Xie, L., Sundaram, H., and Campbell, M. Event Mining in Multimedia Streams. *Proceedings of the IEEE 96*, 4 (Apr. 2008), 623–647.

[243] Yang, J. Toward Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones. In *Proc. IMCE '09* (2009), pp. 1–10.

[244] Zandbergen, P. A. Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. *Transactions in GIS 13*, s1 (June 2009), 5–25.

[245] Zheng, J., and Ni, L. M. An Unsupervised Framework for Sensing Individual and Cluster Behavior Patterns From Human Mobile Data. In *Proc. Ubicomp '12* (2012), pp. 153–162.

[246] Zimmermann, A., Lorenz, A., and Oppermann, R. An Operational Definition of Context. In *Proc. CONTEXT '07* (2007), Springer Berlin / Heidelberg, pp. 558–571.

[247] Zipf, A. User-Adaptive Maps for Location-Based Services (LBS) for Tourism. In *Proc. ENTER '02* (2002).

Appendix A

Chapter 3 Companion

A.1 Calendar Study Consent Form

Calendar Study Consent Form

Tom Lovett

Study Overview

In this study, we will be exploring your calendar use in the office and comparing other data with it – namely basic location data from mobile phones and contacts data from your email account. For the study, we ask that you carry a mobile device with you (we will provide the device) when you are in the office, and for you to keep a written diary of meetings that you attend. A researcher will show you the data that you need to record, but please record all meetings – however spontaneous or informal they may be. There will also be a weekly interview session with a researcher, during which the researcher will go through your calendar entries for the preceding week and compare them with their observations and your diary.

Please be aware: for their observations, the researchers will be monitoring your local team area and recording meeting data throughout the study (see the Data Collection section below). If you are uncomfortable with this, please do not consent to the study.

The study will last for 6 working weeks, and researchers will be on hand throughout for you to ask any questions. You do not need do anything beyond carrying your device, recording your important events in your diary and participating in the weekly interviews. Please try to keep your mobile device charged whenever you can as their batteries are likely to drain quickly.

Data Collection

In addition to your diary data and our observation data, the following data will also be collected and stored during the study:

- Location data: this is a log of the Bluetooth ID of your mobile device over time. We will store the ID along with time records of observations from our static devices placed in your office.
- Email contact data: we will be storing your *personal* (not global) email contacts from your Outlook application on our server. These will be stored anonymously; identified only by a unique alphanumeric contact id. You and other participants in the study will have our uniquely assigned ID linked to this unique ID.
- Calendar event data: we will also be storing your calendar data from your Outlook application on our server. These will be stored exactly as you see them in Outlook, but contacts will be assigned an anonymous unique ID. The event names will be stored in plain text, however.

Your data will be stored anonymously, and will **not** knowingly be shared with any third parties. It will not be used to identify you to anybody. It will not be passed on to anyone within your company who is not involved in this research project, e.g. managers or executives.

Personal Data

ID ....
Age ....
Sex ....
Nationality ....
Role ....

Consent

Name: ....
Signature: ....
Date: ....

Diary Meeting Template

Participant ID:

Meeting Name (leave blank if no name):

Start time (day/month/year hh:mm):

End time (day/month/year hh:mm):

Location (please be as precise as possible):

Attendees:

Appendix B

Chapter 4 Companion

B.1 Transition Study Consent Form

Place study consent form

Tom Lovett

1 Overview

The study is conducted in two parts:

1. A short interview

2. A field study

Both parts are about identifying the moments at which you enter and exit daily places that are meaningful to you. During the interview, the researcher will ask you to describe your typical day in terms of visits to your meaningful places and activities performed while in these places, before asking you to identify a set for you to perform.

During the field study, the researcher will either – depending on the capabilities of your personal mobile device – provide you with a mobile device with a pre-installed application, or install the application on to your personal mobile device. This application will be continuously logging data from a set of motion sensors on the device. The researcher will then ask you to undertake your identified sequence of place visits whilst carrying the mobile device. Each visit must last at least 5 minutes, during which you should perform your identified activity (or activities) as you normally would outside this study. During the study, the researcher will be monitoring you and recording data about your visits; you will be asked to identify the moment in time at which you consider yourself to have entered or exited each place.

2 Information

During the study, the following data will be recorded:

- Details on your activities and places as described by you, and as recorded by the researcher. Only your place labels will be recorded, not their location.

- Details on your motion as recorded by the mobile device motion sensors.

These data will be stored in a pseudo-anonymous format: the only identifying feature will be your initials, and these will only be used by the researcher for identifying data sets. Any results published will be anonymous. The data will *not* be passed to third parties.

If you agree to participate in this study as described, please indicate your agreement by writing your name and email address below, followed by your signature and the date. Thank you for you participation in this research.

Name ....

Email ....

Signature ....

Date ....

B.2 Place Inference Study Consent Form

Place Sequence Consent Form

Tom Lovett

1 Study Overview

This study aims to capture data about your personally meaningful places using a mobile device. A 'meaningful' place is an area which has particular personal meaning to you when you are present at it.

Prior to the study, we will install an application onto your mobile device which will log data throughout the study. You will then be interviewed by a researcher who will ask you to identify a typical week's worth of meaningful places and activities performed within each place *in the sequence that you would visit them.* Once this place sequence has been identified, you will be asked to physically undertake the sequence whilst carrying your device. Just prior to this, you will be required to undertake a short training session (no more than 5 minutes) to calibrate your device.

During the place visits, you will be asked to rate how likely you would be – given your location, activities and surrounding noise levels – to respond to an audio/vibratory notification from your mobile device. This should be rate on a 'high' and 'low' scale, and a researcher will provide you with the equipment to do this.

While you undertake your place visits, we ask that you stay at each place for *at least* 5 minutes and perform the activities that you identify in the interview at each. Please try to carry your mobile device exactly as you would outside the study. A researcher will be following you at a short distance and recording what you do as you go along. You are free to withdraw from the study at any time, and we thank you for your participation.

2 Data Collection

Throughout the study, the device will be collecting various forms of data, namely:

- Time-stamped location data, in the form of latitude-longitude coordinates.
- Place labels specified by you.
- Your sequence of places and activities
- Your rating of how likely you would respond to notifications at any given place.

The location data is stored **locally on the device only**. It is *not* knowingly sent remotely to anyone. After the study, it will be removed from your device by a researcher using a USB cable under your supervision. It will then be purged from your device.

The data is anonymous, i.e. your identity is not stored, and will be identified using a unique number.

3 Demographics

- **ID:**..
- **Age:**..
- **Sex:**..
- **Nationality:**...
- **Occupation:**...

4 Consent

I agree to undertake this study, knowing that I may withdraw at any time.

Name: . . . .
Signature: . . . .
Date: . . . .

Appendix C

Chapter 5 Companion

This appendix contains the consent form for the field study in Chapter 5, as well as raw data on participants' places as collected during study. Section C.1 contains the consent form, and Section C.2 consists of one place table per participant, each containing data on place labels, place sizes and user actions for each place. There is also a column indicating which places were included in the participant test runs.

Section C.3 consists of one distance matrix per participant, each containing data on distances between test places.

C.1 Study Consent Form and Test Instructions

Place Recognition Consent Form

Tom Lovett

1 Study Overview

This study aims to capture and learn about your personally meaningful places using a mobile device. A 'meaningful' place is an area which has particular personal meaning to you when you are present at it.

Prior to the study, we will install an application onto your mobile device which will – throughout the study – attempt to recognise these meaningful places, requiring your help to do so.

Your task, therefore, is to tell the device about these places when you are present in them. The device will sometimes notify you audibly and visually (using the widget on your homescreen) when it requires your feedback, but at times it will silently recognise your places and not notify you.

You can intervene in the device's current decision using a variety of methods, and you can edit the recognised places throughout the study; we will give you a full set of illustrated instructions and a training session on how to do this.

Finally, the application contains a small SMS service that will allow contacts selected by you to receive your device's current or latest place estimate when they send an SMS to you containing various forms of the phrase "Where are you?".

You are free to withdraw from the study at any time. If you do complete the study, you will be paid £20.00 for your time and effort.

2 Data Collection

Throughout the study, the device will be collecting various forms of data, namely:

- Time-stamped location data, in the form of latitude-longitude coordinates.
- Place labels created by you.
- A history of your interventions.

This data is stored **locally on the device only**. It is *not* knowingly sent remotely to anyone. After the study, it will be removed from your device by a researcher using a USB cable under your supervision. It will then be purged from your device.

The data is anonymous, i.e. your identity is not stored, and will be identified using a unique number.

3 Demographics

- **ID:**...
- **Age:**...
- **Sex:**..
- **Nationality:**..
- **Occupation:**..

4 Consent

I agree to undertake this study, knowing that I may withdraw at any time. I also acknowledge that I will be paid £20.00 if I complete the study in full.

Name: . . . .
Signature: . . . .
Date: . . . .

Place Testing Instructions

Tom Lovett

1 Test Overview

Thank you for completing the first part of this study. For this part, please install a test application that will be provided to you by a researcher. The application is simple, and the researcher will show you how to use it.

For testing:

- Please think of a few meaningful places that are within walking distance from you now. Once you have done this, please write them down on a separate sheet of paper provided to you by a researcher.
- Next, please identify a natural path between these places. You do not need to visit a place more than once.
- Next, please write down a list of places along the identified path that are **not** meaningful to you.
- Now, you will be asked to follow your identified route and at each place along this route – both meaningful and not meaningful – you will be asked to open the test application on your mobile phone, select the place (or 'non-place') from the menu, and press the 'Sample' button. **Each time you press 'Sample'**, place the device in a natural position according to how you would normally carry it in each spot, e.g. in your pocket or a bag.

  Please do this 5–10 times in each place but **do not** take samples whilst standing at the same spot; rather, move to separate locations within each place before pressing 'Sample' again. These separate locations should be plausible, i.e. locations that you would likely visit in each place, not random locations.

Once you have finished your route, return to the researcher. The test application and its data will then be removed from your device. The original application used in the first part of the study will also be removed. All data related to this study will be purged from the device.

Once done, you will be asked to fill in a short survey on your study experience. Thank you again for your participation.

2 Demographics

- **ID:**. . . . .

C.2 Participants' Places

	Label	Tested	CONFIRMED	CORRECTED	DELETED	RELABELLED	SIZE
1	Desk	1	29	14	0	0	<ROOM
2	Kitchen	1	0	3	0	0	ROOM
3	Costa	1	1	2	0	0	ROOM
4	Home	0	48	0	0	1	BUILDING
5	Home upstairs	0	0	5	1	0	ROOM
6	Alex's house	0	8	0	0	0	BUILDING
7	Pitstop	0	0	0	0	0	ROOM
8	Music room	1	11	0	0	0	ROOM
9	Sainsbury's	0	0	1	0	0	BUILDING
10	Claverton rooms	1	5	0	0	0	ROOM
11	Tiki	0	2	0	0	0	ROOM
12	Kate's house	0	27	0	0	0	BUILDING
13	Town	0	11	0	0	0	>BUILDING
14	ICIA	1	2	0	0	0	ROOM
15	SU upstairs	1	0	0	0	0	ROOM
16	Balcony	1	1	0	0	0	<ROOM
17	Parade	0	4	3	0	0	BUILDING

	Label	Tested	CONFIRMED	CORRECTED	DELETED	RELABELLED	RESET	SIZE
1	Desk	1	16	57	0	0	2	<ROOM
2	Procrastination chairs	1	13	14	0	0	3	<ROOM
3	HCI Lab	1	20	48	0	0	2	ROOM
4	Costa	1	10	1	0	0	0	ROOM
5	Loo	0	0	0	1	0	0	ROOM
6	Kitchen	0	0	2	1	0	0	ROOM
7	Bus stop - uni	1	4	0	0	0	0	ROOM
8	Home	1	67	0	0	0	0	BUILDING
9	Bus stop - home	0	5	0	0	0	0	ROOM
10	Fresh	1	3	0	0	0	0	BUILDING
11	Kitchen	0	5	13	0	1	0	ROOM
12	Gym	0	12	1	0	0	0	BUILDING
13	Town	0	12	0	0	0	0	>BUILDING
14	Balcony	0	0	0	1	0	0	<ROOM
15	Parade bar	0	4	0	0	0	0	ROOM
16	balcony	1	0	3	0	0	0	<ROOM
17	bath spa station	0	1	1	0	0	0	BUILDING

	Label	Tested	CONFIRMED	CORRECTED	DELETED	RELABELLED	RESET	SIZE
1	work.working	1	18	37	0	1	1	<ROOM
2	work.procrastinating chairs	0	0	0	1	1	0	<ROOM
3	on the bus home	0	4	0	1	1	0	<ROOM
4	sainsburys town	0	4	0	0	0	0	BUILDING
5	home	0	88	0	0	0	0	BUILDING
6	Med center	1	0	1	0	0	0	BUILDING
7	hci lab	0	0	2	1	0	0	ROOM
8	su	0	1	0	0	0	0	BUILDING
9	moorland road	0	6	0	0	0	0	>BUILDING
10	bus stop for home to uni	0	2	0	0	0	0	ROOM
11	uni bus stop	1	2	0	0	0	0	ROOM
12	Tkd dojo	0	2	0	0	0	0	ROOM
13	Victoria park	0	0	0	0	0	0	>BUILDING

	Label	Tested	CONFIRMED	CORRECTED	RESET	SIZE
1	Kitchen	0	0	3	0	ROOM
2	Desk	1	16	26	0	<ROOM
3	Uni bus stop	1	2	3	1	ROOM
4	Home	0	28	1	0	BUILDING
5	Stv	1	3	0	0	BUILDING
6	Hci lab	1	0	9	0	ROOM
7	Morrison's	0	0	0	0	BUILDING
8	Swindon	0	0	0	0	>BUILDING
9	London	0	2	0	0	>BUILDING
10	City's Bus stop	0	0	0	0	ROOM
11	Green outside office	1	0	0	0	ROOM
12	Toilet	1	0	0	0	ROOM
13	Jjb bus stop	0	0	0	0	ROOM

	Label	Tested	CONFIRMED	CORRECTED	SIZE
1	desk	1	66	1	<ROOM
2	breakout	1	0	3	<ROOM
3	printers	1	1	7	<ROOM
4	RnD room 2	1	0	1	ROOM
5	kitchen	0	0	2	<ROOM
6	bike shed	1	0	0	ROOM
7	home	0	15	0	BUILDING
8	coffee shop	1	2	0	ROOM
9	Baird house mtg room	1	0	0	BUILDING
10	Nby train station	0	1	0	BUILDING
11	vf Paddington	0	8	0	BUILDING
12	Restaurant	1	3	0	ROOM
13	nby Waitrose	0	0	0	BUILDING
14	one stop	0	1	0	BUILDING
15	sun in wood. pub	0	7	0	BUILDING
16	Babbage house	1	0	0	BUILDING
17	ground floor breakout	1	0	0	<ROOM

	Label	Tested	CONFIRMED	CORRECTED	RELABELLED	SIZE
1	Desk	1	6	0	0	<ROOM
2	Emma's House	0	0	0	0	BUILDING
3	Home	0	5	1	0	BUILDING
4	Coop	0	0	0	0	BUILDING
5	Beautician	0	0	0	0	BUILDING
6	STV	1	2	0	0	BUILDING
7	library	1	0	0	0	BUILDING
8	Costa SU	1	0	0	0	ROOM
9	World's best coffee shop	0	0	0	1	ROOM
10	Kai's Flat	0	27	0	0	BUILDING
11	Peckham Rye Station	0	3	0	0	BUILDING
12	Hyde Park	0	1	0	0	>BUILDING
13	Olympic Park	0	1	0	0	>BUILDING
14	Clapham Junction Station	0	1	0	0	BUILDING
15	Eton Dorney Canoe Sprint Venue	0	0	0	1	>BUILDING
16	Windsor + Eton Riverside	0	0	0	1	>BUILDING
17	Roskilde Hotel	0	5	0	0	BUILDING
18	Roskilde University PDC2012	0	1	0	0	BUILDING
19	Copenhagen airport	0	1	0	0	BUILDING
20	Paddington Station	0	0	0	0	BUILDING

	Label	Tested	CONFIRMED	CORRECTED	RELABELLED	SIZE
1	desk at work	1	23	28	0	<ROOM
2	Costa coffee	1	3	4	0	ROOM
3	home	0	30	0	0	BUILDING
4	bus stop in town	0	3	1	0	ROOM
5	HCI Lab	0	0	0	0	ROOM
6	SU Shop	1	0	0	0	ROOM
7	uni bis stop	1	1	2	0	ROOM
8	tiki coffee	0	2	0	0	ROOM
9	balcony seats	0	0	2	0	<ROOM
10	Southgate shopping...	0	1	0	0	BUILDING
11	small park	0	0	0	0	>BUILDING
12	Starbucks next to bus stop	0	0	0	0	ROOM
13	Tesco	0	2	0	0	>BUILDING
14	East Building	1	4	6	1	BUILDING
15	east building chairs by kitchen	0	0	0	0	<ROOM
16	mums house	0	10	0	0	BUILDING
17	Greek market maidstone	0	1	0	0	>BUILDING
18	maidstone	0	0	0	0	>BUILDING
19	Nero maidstone	0	0	0	0	BUILDING

	Label	Tested	CONFIRMED	CORRECTED	SIZE
1	Vodafone meeting tables	1	0	0	<ROOM
2	Tom's desk	1	2	1	<ROOM
3	Kevs desk	1	0	7	<ROOM
4	VF canteen outside	1	0	0	ROOM
5	Home	0	41	0	BUILDING
6	Sanctus drive	0	7	0	ROOM
7	Thatcham bus stop 20	0	2	0	ROOM
8	VF bus stop	1	2	0	ROOM
9	Printer	1	0	0	<ROOM

	Label	Tested	CONFIRMED	CORRECTED	SIZE
1	Home	1	34	0	ROOM
2	Post Office	0	0	0	BUILDING
3	Work	1	17	0	BUILDING
4	Work Car Park	1	0	1	BUILDING
5	Waterloo Car Park	1	2	2	BUILDING
6	Wedding	0	0	0	ROOM
7	Ant's House	0	0	0	BUILDING
8	Storyteller's	1	0	0	ROOM
9	Wedding reception	0	0	0	BUILDING
10	M&S	1	0	0	BUILDING
11	Fat face	1	0	0	ROOM
12	Tom's Parents	0	15	0	BUILDING
13	Coffee no.1	0	0	0	BUILDING

	Label	Tested	CONFIRMED	CORRECTED	DELETED	RESET	SIZE
1	Home	0	19	0	1	4	BUILDING
2	Home	1	52	0	0	0	BUILDING
3	Starbucks	0	1	0	0	0	BUILDING
4	Desk	1	30	5	0	0	<ROOM
5	Kitchen	1	0	2	0	0	ROOM
6	Chairs	1	0	2	0	0	<ROOM
7	Fresh	1	1	0	0	0	ROOM
8	Library	1	0	0	0	0	BUILDING
9	Costa	1	1	0	0	0	ROOM
10	Meeting area	1	0	0	0	0	<ROOM
11	Desk	1	11	0	0	1	<ROOM
12	Printer	0	0	1	0	0	<ROOM
13	South tables	0	0	0	0	0	<ROOM
14	Kitchen	0	1	2	0	1	<ROOM
15	Wedding	0	0	0	0	0	BUILDING
16	Daphne	0	0	0	0	0	BUILDING
17	Ant's	0	0	0	0	0	BUILDING
18	Storyteller	0	0	0	0	0	BUILDING
19	Wedding reception	0	0	0	0	0	>BUILDING
20	M and S	0	0	0	0	0	BUILDING
21	Finance	1	0	0	0	0	BUILDING
22	Parents	0	2	0	0	0	BUILDING
23	Red Lantern	0	1	0	0	0	BUILDING

C.3 Test Place Distance Matrices

The following matrices contain approximate distances in metres between test places (one matrix per participant).

	Desk	Kitchen	Costa	Music.room	Claverton.rooms	ICIA	SU.upstairs	Balcony
Desk	0	160	220	340	400	300	220	0
Kitchen		0	360	500	540	440	360	40
Costa			0	140	180	80	0	360
Music room				0	60	80	140	500
Claverton rooms					0	100	180	540
ICIA						0	80	440
SU upstairs							0	340
Balcony								0

	Desk	Procrastination.chairs	HCI.Lab	Costa.	Bus.stop...uni.	Home	Fresh	balcony
Desk	0	20	20	360	180	2840	680	0
Procrastination chairs		0	20	400	200	2840	700	0
HCI Lab			0	380	180	2820	680	20
Costa				0	180	2620	320	380
Bus stop - uni					0	2700	500	200
Home						0	2460	2840
Fresh							0	700
balcony								0

	work.working.	Med.center.	uni.bus.stop.
work.working	0	640	60
Med center		0	640
uni bus stop			0

	Desk	Uni.bus.stop	Stv	Hci.lab	Green.outside.office	Toilet
Desk	0	140	100	0	360	20
Uni bus stop		0	160	140	500	160
Stv			0	100	420	120
Hci lab				0	360	20
Green outside office					0	340
Toilet						0

	desk	breakout	printers	RnD.room.2	bike.shed	coffee.shop	Baird.house.mtg.room	Restaurant	Babbage.house	ground.floor.breakout
desk	0	20	20	20	80	120	140	80	60	20
breakout		0	0	0	100	100	120	80	60	0
printers			0	0	80	100	120	80	40	20
RnD room 2				0	80	100	120	80	60	20
bike shed					0	200	180	160	60	100
coffee shop						0	80	40	140	100
Baird house mtg room							0	80	120	120
Restaurant								0	100	60
Babbage house									0	60
ground floor breakout										0

	Desk	STV	library	Costa.SU
Desk	0	180	380	260
STV		0	460	380
library			0	60
Costa SU				0

	desk.at.work	Costa.coffee	SU.Shop	uni.bis.stop	East.Building
desk at work	0	260	280	100	80
Costa coffee		0	20	160	340
SU Shop			0	180	360
uni bis stop				0	180
East Building					0

	Vodafone.meeting.tables	Tom.s.desk	Kevs.desk	VF.canteen.outside	VF.bus.stop	Printer
Vodafone meeting tables	0	500	480	460	540	460
Tom's desk		0	20	60	80	40
Kevs desk			0	60	80	20
VF canteen outside				0	80	80
VF bus stop					0	100
Printer						0

	Home	Work	Work.Car.Park	Waterloo.Car.Park	Storyteller.s	M.S	Fat.face
Home	0	27060	26820	100	24180	300	720
Work		0	440	26960	3360	26920	26660
Work Car Park			0	26740	3360	26700	26440
Waterloo Car Park				0	24080	220	620
Storyteller's					0	24040	23740
M&S						0	420
Fat face							0

	Home	Desk	Kitchen	Chairs	Fresh	Library	Costa	Meeting.area	Desk	Finance
Home	0	55040	55040	55040	55460	55300	55260	55020	78760	55500
Desk		0	20	20	680	440	360	40	111620	700
Kitchen			0	0	680	440	360	40	111640	700
Chairs				0	680	440	360	40	111640	700
Fresh					0	240	320	660	112300	40
Library						0	80	440	112060	260
Costa							0	360	111980	340
Meeting.area								0	111640	680
Desk									0	112320
Finance										0

C.4 Raw Location Accuracy Data

Figure C.1 shows the raw location accuracy data for the participants during field deployment, grouped by location provider.

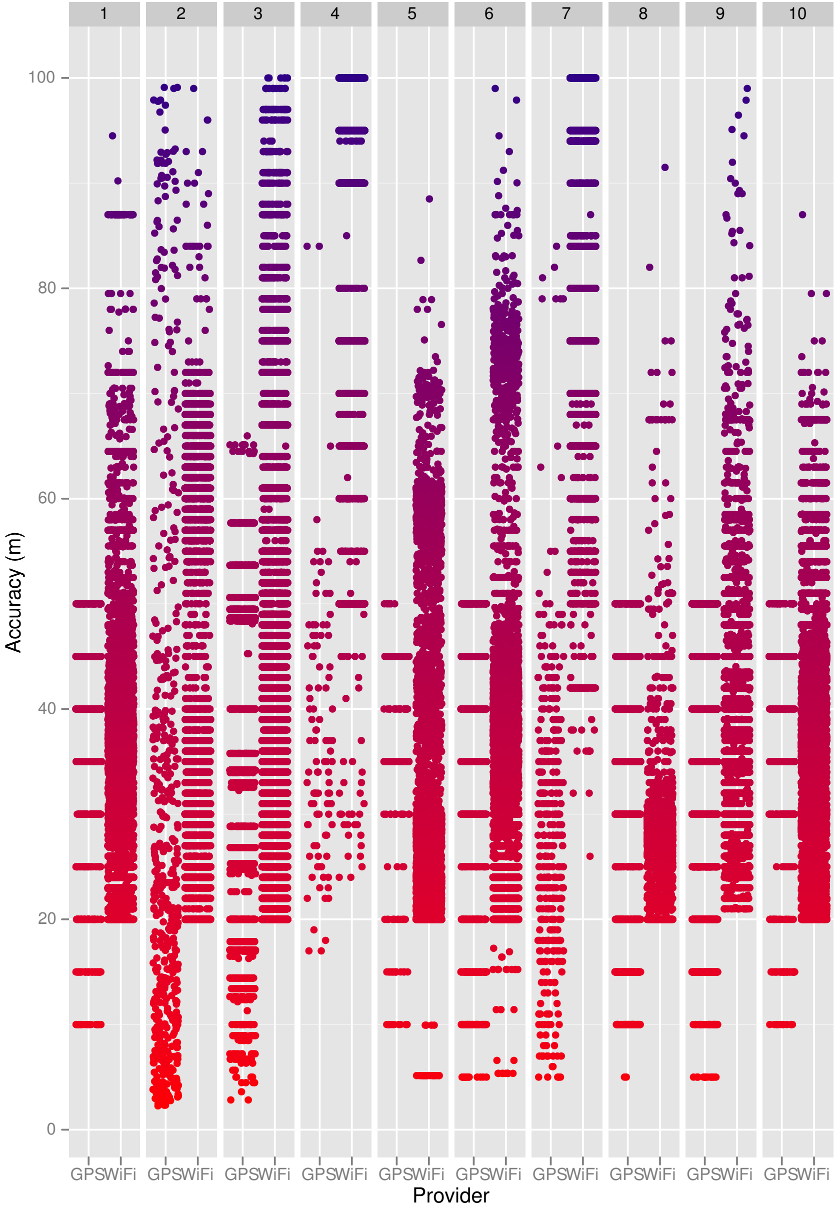


Figure C.1: *Location accuracy, i.e. metres radius, of all location samples over the participants and location sensors.*