A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the School of Information Technologies at The University of Sydney

William Tse-Yun Niu
March 2009
© Copyright by William Tse-Yun Niu 2009
All Rights Reserved
I have examined this thesis and attest that it is in a form suitable for examination for the degree of Doctor of Philosophy.

(Judy Kay)  Principal Advisor
for ah-ma and my parents
Acknowledgements

First of all, I would like to express my deep appreciation to my supervisor, Judy Kay. I could not have gone through this degree without her incredible guidance, patience, intelligence, and thoughtfulness. She has given much more than what a Ph.D. student should expect from a supervisor, in terms of her time, efforts, and care.

Next, I would like to thank my colleagues whom I have established lasting friendship with, especially the G61b-ers: Trent Apted, Mark Assad, Michael Avery, David Carmichael, Derek Corbett, Daniel Cutting, Adam Hudson, Daren Ler, Andrew Lum, David Storey, and David Symonds. G61b-ers made this process much more fun and smooth than it otherwise would.

Of those, I especially want to thank some of them: Trent for his in-depth knowledge in Linux and various other things, including Lyx, coffee, chocolate, Australian slang, beers...; David (Carmichael) for his help on various projects and being a great companion socially and academically; Adam for always being patient in giving instructions and advice; and Andrew for handing down much knowledge—for having related thesis topics—and sharing great trips in Europe. I would also like to thank other colleagues who enabled me to shake off stress in the process and to share ideas: Anthony Collins, Greg Darke, Lichao (Bruce) Li, Meena Rajani, Rajibussalam, and Jorge Villalon.

Cheng (Gary) Weng, whom I stayed with in Sydney throughout my degree, has been a great mate. Having shared a lot of commonalities and interests, we have gone through many ups and downs in the past six years.

I want to thank Carolin Plate for her technical writing skills, knowledge about running user studies, encouragement, understanding, and, best of all, being a great friend. One of the best things that has happened to me during my Ph.D. is getting to know her.

There are also many other friends who emotionally and technically supported me, in particular Kai-Lo (Carol) Ling, Ta-Yuan (David) Wang, Coco Hsu, Iris Wong, Yi-Hsing Chen, Frances Kuo and Bun Chan. I thank them for being there for me.

Then, I would like to thank those who reviewed various parts of my thesis: Anthony
Collins, Dan Cutting, Adam Hudson, Michael Kruppa, Andrew Lum, and Carolin Plate. Their valuable comments made this thesis more presentable.

I would like to thank Smart Internet Technologies CRC and the university/faculty for their generous financial support.

Lastly, I want to thank my parents and my grandmother for their most candid care and support. In addition, I would also like to thank my extended family, especially my uncle/god dad, Frank Hsu, who is always at my back giving me just the support I needed.
Publications

Journal Article

Conference Proceedings


Workshop Proceedings

2. Judy Kay and William Niu. Adapting information delivery to groups of people. In Workshop on New Technologies for Personalized Information Access at UM ’05,


Technical Reports


Abstract

One challenge in pervasive computing is to effectively model and reason about the environment and entities within it, in order to provide context-aware adaptation. This thesis explores new ways to improve reasoning in pervasive computing by exploiting the power of a personal ontology. This goes beyond the body of work that has primarily focused on reasoning complexity and performance of ontological reasoning about context-aware computing. We have created Personalised Pervasive Scrutable Ontological Framework (PERSONAF), a conceptualisation and implemented framework for ontological reasoning about people’s locations within an indoor pervasive computing environment. We demonstrate how PERSONAF addresses three critical challenges for pervasive computing systems: conflict resolution for sensor fusion; personalised information delivery; and explanation of personalisation.

A key contribution of this work is its support for personal ontologies as a basis for presenting personalised information within a pervasive computing environment: one concept may have different meanings to different people under different situations. So, for example, Room 125 in a building may be a social hub for one person, a recharging corner for some people, a coffee room for some others or an ordinary common room for others. The ontological knowledge base of the PERSONAF framework is capable of capturing this kind of context-dependent propositions. This thesis tackles the important issue of how to construct such personal ontologies in a way that avoid undue load on users. We semi-automatically populate a base ontology with propositions extracted from multiple domain-relevant sources. Evidence for each proposition found is preserved for ontological reasoning later.

There are two main components in the PERSONAF framework: a central ontological knowledge base—the Personalised Context Ontology (PECO)—and a reasoning component—the Ontology- and Evidence-based Context Reasoner (ONCOR). PECO consists of three layers: a middle ontology that captures the ontological model for a generic building and this may be integrated with or linked to a top-level ontology; an
application ontology which is constructed from a set of building maps to model a particular building, in our case, the School of Information Technologies building on our campus; and a personalised and light-weight layer of the accretion ontology. The accretion ontology accumulates propositions extracted from various sources and store them as evidence. PECO holds all the ontological information and then, when an application needs to access the information, the Value Resolution process is performed by ONCOR.

ONCOR is a reasoning mechanism especially designed for interpreting PECO. It performs the resolution process of the accretion and resolution approach in an ontological manner: when a value is required for a concept in a proposition (e.g. where is Bob?), the stored evidence is examined and resolved to a value. This process can be personalised depending on the context of the program or person requiring the value. This is done by running a customised resolver function.

This thesis describes two experiments in order to evaluate the PERSONAF framework:

- The results of the location conflict resolution study illustrate that ontological reasoning was able to drop the mean error rates from 55% with a commonly naive algorithm to 16% with the best of the ontologically based algorithms. This work provides the first implementation of such an approach with a range of ontological reasoning algorithms explored and evaluated.

- In another study aiming to evaluate an adaptive system back-ended with PECO and ONCOR, the results indicate that people preferred the adaptive and scrutable system over the non-adaptive one even though they could complete the designed tasks with both systems.

In summary, the thesis tackles three key problems for pervasive computing. Firstly, resolve conflict in terms of fusing location sensor data, addressing challenges in location modelling, an essential element of context-aware systems. Secondly, we personalise pervasive information to users, pursuing a key aspect of Weiser’s driving vision of pervasive computing. Thirdly, we generate comprehensible explanations of personalisation, an issue acknowledged as critical in pervasive adaptive systems. We have addresses these problems through creating, implementing, and evaluating the PERSONAF framework.
## Contents

Acknowledgements vi

Publications viii

Abstract xi

List of Tables xvi

List of Figures xix

1 Introduction 1
   1.1 Scenario 3
   1.2 Challenges 7
   1.3 Contributions and Thesis Outline 12

2 Background 14
   2.1 Ontologies for Pervasive Computing 14
   2.2 Personal Ontology and Personalised Location Labels 27
   2.3 Location Conflict Resolution 30
   2.4 Personis and PersonisAM 31
   2.5 Summary 33

3 The PERSONAF Conceptual Framework 34
   3.1 Middle Ontology 36
   3.2 Application Ontology 37
   3.3 Accretion Ontology 38
   3.4 Reasoning with PECO: ONCOR 38
      3.4.1 Location Resolvers 39
      3.4.2 Other Resolvers 44
   3.5 Summary 46
4 Using PERSONAF in Pervasive Computing

4.1 Middle Building Ontology
   4.1.1 Ontology Representation
   4.1.2 Reusing Existing Ontologies

4.2 Application Ontology

4.3 Accretion Ontology
   4.3.1 Stereotypes
   4.3.2 Facts
   4.3.3 Observations
   4.3.4 Given Information
   4.3.5 Inferences
   4.3.6 Discussions

4.4 Preliminary Evaluation of PECO
   4.4.1 Syntactic Validation
   4.4.2 Scenario- and Comparison-based Evaluation
   4.4.3 Discussion

4.5 Summary

5 Location Conflict Resolution

5.1 Experimental Procedure

5.2 System Design
   5.2.1 Location Log (LocLog)
   5.2.2 Location Log Feedback

5.3 Sensor Infrastructure

5.4 Analysis on User Data and Sensor Data
   5.4.1 Participant Feedback and Self-assessment on LocLog Entries
   5.4.2 User-logged Data
   5.4.3 Sensor Data
   5.4.4 Resolving Location Values

5.5 Summary

6 Pervasive Personalisation Reasoning

6.1 System Overview

6.2 The User Study
   6.2.1 Participants
   6.2.2 Systems
   6.2.3 Data Collection
List of Tables

2.1 An sample list of location sensor evidence (top) and resolved values from three different resolvers (bottom) .................................................. 32

3.1 An example list of sensor evidence ........................................... 40

4.1 A sample view of extracted technical building data ..................... 55

5.1 A summary on reliability questions answered by each participants. Answer are from 1–5, ranging from most reliable (roughly 90%+ accuracy) and least reliable (roughly less than 25% accuracy) .................... 79

5.2 A summary of user-logged data .............................................. 80

5.3 All possible combinations for any two consecutive user-logged messages 81

5.4 Classification of sensor evidence in percentage of time .................. 85

6.1 Time of completion of the two tasks for the two systems. Data are in means ± standard deviations. ...................................................... 101

6.2 Questions and responses of the post-study questionnaire ............... 102
# List of Figures

1.1 An example of Adaptive Locator’s view for two users .............................. 4
1.2 An example explanation of location resolution by reasoning with a spatial ontology ........................................................................................................... 11

2.1 Part of the entity hierarchy of GAIA, reproduced according to the image in (Page 26 of McGrath et al. 2003) ......................................................... 17
2.2 The structure layout for COBRA-ONT v0.4, reproduced according to the image in (Page 180 of Chen et al. 2004c) .................................................... 18
2.3 A upper-level context ontology and two extended ontologies, from (Page 34 of Wang et al. 2004a) .............................................................. 21
2.4 Spatial Representation in UbisWorld in the Ontology Browser on ubis-world.org (accessed 23 August 2006) .............................................................. 23
2.5 GAS Core Ontology, reproduced according to the image in (Page 268 of Christopoulou et al. 2005) .......................................................... 25
2.6 Ontologies in ec(h)o, from (Page 8 of Hatala et al. 2005) ......................... 26

3.1 The PERSONAF framework ................................................................... 35
3.2 The ONCOR component of the PERSONAF framework in Figure 3.1 ... 39
3.3 An example explanation of location resolution using the Granularity Harmoniser approach ............................................................... 43
3.4 An example of MyPlace’s view for two users (duplicate of Figure 1.1) .... 46

4.1 A snippet of a personalised context ontology ................................. 49
4.2 A sample of the building ontology extending MIBO (top). The full building ontology consists of a total of 573 location instances ......................... 52
4.3 Maps for level 1 in the SIT building as building plans and XML/SVG graphics .................................................................................................................. 53
4.4 Possible situations of adjacent locations ................................. 57
4.5 The user may specify their current location using a map-based system 60
4.6 The Web interface for registering a new Bluetooth device to the Locator system .................................................. 61

5.1 Photos of offices (left) and open-area personal work spaces (right) in the SIT building .................................................. 70

5.2 A Tablet PC was used for location logging inside the building ........ 71

5.3 A screenshot of LocLog when a user niu, who knows Bob, clicks on Room 324 on the map at the “Level 3” tab. .......................... 72

5.4 A message box to the top right-hand corner displays the information to be logged, unless the user decides to undo this action. ............. 73

5.5 A screenshot after the user indicates their arrival at Room 324 when the network connectivity is unstable. .......................... 74

5.6 The Location Log Feedback system. The table to the left, the LocLog table, shows user-logged entries: (a) is the ID column that numbers each entry; (b) shows time of each logged entry as well as the gap between this one and its preceding entry; (c) shows the action and purpose, if given, of the entry; (d) is the location logged; (e) is for users to give comments for each entry; (f) is the duration for the day; (g) shows the two highlighted rows that are potentially erroneous; (h) is a basic time calculator for users to better estimate the quality of the logged entries; (i) contains the two self-assessment questions to be completed at the end of the day, and (j) is the area where the user puts more comments about the entries. The table to the right, the Sensor table, lists the evidence sources from sensors: (1) is the ID column; (2) is the timestamp of the evidence source; (3) is the sensor message; (4) lists the inferred location as well as the sensor type (i.e. BT for Bluetooth sensors, Sys for activity sensors and Login for login sensors), and (5) shows the highlighted rows, temporally close to the 8th logged entry clicked by the user; it highlights the rows that are less than 2 minutes away (i.e. 2 minutes backward and forward) from the clicked entry with a darker green colour, and those that are less than 20 minutes away with a lighter green colour. .......................... 76

5.7 Sensors used in the SIT building for the LocLog study: Bluetooth sensors are shown as white circles, activity sensors are shown as black circles, login sensors are actively scanning on computers in the Level 1 labs shown as green squares. .......................... 78
5.8 Classifications of the participants’ behaviour in terms of entries (a) and
time (b) ................................................................. 83
5.9 Classifications of the participants’ behaviour from logged data for each
participant ............................................................... 84
5.10 One day of logged and sensor data for participant E ...................... 86
5.11 An overview of both logged and sensor data for the most and the second
least mobile participants ............................................ 87
5.12 Error rates for different resolvers reasoning at Room/Desk grain-size . 89
5.13 Error rates for different resolvers reasoning at MultiRoomUnit grain-size 90

6.1 A screen shot of the Adaptive Locator as seen by the user marked current
user ................................................................. 94
6.2 The pop-up explaining the darkness of the colour .............................. 95
6.3 The expanded version of the explanations for showing Ian’s location as
“his desk” ........................................................... 96
6.4 The user progressively worked through the tasks on the left screen as they
used the system on the right screen. ................................... 97
6.5 A map-view of the work spaces for the participants ......................... 98
6.6 The non-adaptive version of the location system .............................. 99
6.7 The actual and user perceived system mistakes ................................ 101
Chapter 1

Introduction

Ontologies offer great potential in supporting personalised and contextualised reasoning in pervasive computing\(^1\). This thesis explores ways to create a framework for ontological reasoning to support personalisation and location modelling, enhancing the MyPlace system (Carmichael et al. 2005) which can make use of information from sensors to model the devices and people in an indoor pervasive computing environments. In particular it tackles three critical issues: conflict resolution in location sensor fusion, personalised location information delivery, and explanation of personalisation in adaptive systems.

The seminal work of Weiser (1993) depicts ubiquitous computing as a paradigm of computing that strives for unobtrusive, low-cost, and distributed computing units that are seamlessly deployed in the physical environment\(^2\). It is also known as pervasive computing, ambient intelligence, and everyware. In this thesis, we refer to this area of research as pervasive computing.

In the spirit of pervasive computing, a smart building should be able to provide the types of information that a human concierge does. So, for example, suppose Yvonney enters the building. The concierge may immediately volunteer the information that Wendy, her close friend and coworker, is probably in the common room as she headed that way a few minutes ago, carrying her coffee cup. An hour later, an important new client, Zack enters the building for a meeting with Wendy. The concierge tells him to

\(^1\)For example, Coutaz et al. (2003); Ranganathan and Campbell (2003b); Chen et al. (2004a); Wang et al. (2004b); Christopoulou and Kameas (2005); Hatala and Wakkary (2005); Heckmann et al. (2005); Wishart et al. (2005); Masuoka et al. (2003); Strang et al. (2003a); Becker and Nicklas (2004); Elenius and Ingmarsson (2004); Gandon and Sadeh (2004); Korpiïaï et al. (2004); Wang et al. (2004a); Xu et al. (2004); de Freitas Bulcã£o Neto and da Graça Campos Pimentel (2005); Lin et al. (2005); Tan et al. (2005); Agostini et al. (2006); Specht and Weithoner (2006); Strimpakou et al. (2006); Weißenberg et al. (2006); Zhu and Madnick (2006); Ejigu et al. (2007).

\(^2\)Other early significant contributions in this area of research also include Sakamura (1987); Want, Hopper, Falcão, and Gibbons (1992); Lamming and Flynn (1994); Schilit, Adams, and Want (1994); Negroponte (1996).
go to Room 802 (Wendy’s office) and directs him to the appropriate lifts.

We can usefully distinguish three levels of knowledge that are needed for provision of the ontological reasoning to support personalised information about a building.

1. Firstly, a concierge needs commonsense knowledge of buildings in general. At this level, they must know about the concepts, floor, room, wing, foyer and the like.

2. Secondly, a good concierge would acquire a very sound knowledge of this building. So, they would know that Room 801 is a 200-metre walk from Room 802 even though they have a common wall: it involves walking along a corridor, through a security system and another long corridor. They would also know that Wendy’s office is Room 802 and how to get there easily.

3. A third level of knowledge is far more dynamic, including knowledge about the people in the building and relationships between them. This layer involves the continually changing knowledge about the location of people in the building.

To make use of this knowledge, the MyPlace concierge must also be able to reason about each of these layers of their knowledge. So, the concierge would be able to reason that if they saw Yvonney walking towards the coffee room one minute ago, she is likely to still be there. But after an hour, that information is much less useful for inferring her current location. Similarly, the concierge would reason that the workers using the offices in Rooms 801 and 802 would not have much contact, in the building layout described above. However, where Rooms 802 (Wendy’s office) and 803 (Yvonney’s office) have a shared entry, they are likely to know each other well. So it would be natural for the concierge to refer to Yvonney by name when speaking to Wendy.

A very important part of this reasoning is the personalisation that the concierge can do based on the knowledge of each person’s understanding, their personal ontologies. This notion might, at first, seem to conflict with the usual definitions of ontologies, where the emphasis is on shared and reusable understandings. For example, two influential definitions for ontology are “an explicit specification of a conceptualisation” by Gruber (1993) and “a shared understanding of some domain of interest” by Uschold and Grüninger (1996). Many researchers use a combination of those two: an explicit specification of a conceptualisation that is shared within some domain of interest. This captures the spirit of much research in ontologies, with its focus on the shared understanding that all people have about one domain.

For the MyPlace vision, we need to go beyond this to capture the particular conceptualisation of each user of the building. More than that, we need to be able to make use
of the conceptualisation that each user is willing to make available to each other user. So, for example, for a newcomer to the building, like Zack, MyPlace should infer that he can understand how to use the concept Room 802. However, for Yvonney it would be better to refer to it as Wendy’s office. Similarly, if the Board Room is used for a Leadership Course, a person who comes to the building only for that course would think of that room as the Leadership Course Classroom and that would be a very good way to describe it for them.

A more subtle form of personal ontology concerns the relationships between the people in the building. For example, as Yvonney and Wendy are good friends, Yvonney may be happy if the MyPlace system tells Wendy precisely where she is in the building. She may also be happy for Wendy to be told where she was an hour ago, especially if that is the best available information. By contrast, Yvonney may prefer that her boss knows only whether she is in the building right now, not her precise location and not the details of her previous locations. Ontological knowledge of the building supports the reasoning needed for this.

This thesis describes our work to create a framework for representing and reasoning about a new form of ontology that can provide the foundation for a pervasive computing system such as this vision for MyPlace. It can enable the MyPlace interface to present information about the building in a form that is personalised on the basis of a model of the user and models of other people in the building. For this, we extend the definitions already given for ontologies to: an explicit specification of a conceptualisation, which adapts to the context and users. Here we draw on the notion of context, which has been variously defined but one influential definition is “Context is 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 applications themselves” (Dey 2001). This highlights the need to support flexible reasoning about the representation of the building conceptualisation so that MyPlace can adapt reasoning to the newest and most relevant information that is available about the current context. This dynamic character is an essential characteristic of pervasive, or context-aware computing.

1.1 Scenario

Figure 1.1 illustrates the types of personalisation that are supported in Adaptive Locator user interfaces, which provide information about people’s locations. In this example, we suppose that two users, Alice and Bob, consult their own instances of the Adaptive
CHAPTER 1. INTRODUCTION

Alice's view
Carol is in Coffee Room
David is in Room 300

Bob's view
Carol is in SIT Building
David is in his office
Eva is in Room 310

Figure 1.1: An example of Adaptive Locator’s view for two users

Locator interface to find out who is in the School of Information Technologies (SIT) building. Adaptive Locator uses a personalised context ontology to capture the contexts and the understanding of each user in a pervasive computing environment. This gives the different selection of information for each of them and different ways to express the same information.

Location Granularity

Alice is presented with finer-grained location information about Carol than Bob, assuming the coffee room is in the SIT building. The system could have presented the location as Room 125 or, at an extreme precision, but unfriendly information, (-33° 53’ 17.88”, +151° 11’ 38.40”) as geographic coordinates in latitude and longitude. The unnaturalness of geographic coordinates usually makes it unacceptable to present to end users. Ashbrook and Starner (2003) and Hightower (2003) argue that location labels need to be more human-centric and user-friendly than geographical coordinates for pervasive computing applications, and a user study presented by Weilenmann and Leuchovius (2004) confirmed this argument. Zhou et al. (2005) has pointed out the importance of people’s use of different descriptions for the same places, such as “the coffee shop”, “Starbucks”, and “the place we met last time”.

There are several possibilities in addressing this difference in location presentation, is this case, coffee room and SIT building.

1. One is that a building-level of granularity is what Bob needs to know about Carol’s location, and therefore, Bob chooses an appropriate location resolution algorithm to do so. For instance, Bob might want to see whether Carol is currently in the building for a quick meeting. This reasoning can be supported by a spatial ontology with containment relations, capturing the notion that the coffee room is-part-of the SIT building. The Adaptive Locator system may be enabled to allow the user
1.1. SCENARIO

to select a location resolution algorithm and to display coarser levels of location granularity.

2. One other possibility for the difference is that the system adaptively chooses different location resolution algorithms to resolve Carol’s location for Alice and Bob. If it is 10 minutes past the time of a meeting scheduled between Bob and Carol, the system may infer Bob’s query about Carol’s location is to find her for the meeting. In that case, a building-level of granularity may suffice the query. This may require the system to infer (e.g. by learning) the most suitable algorithm for each context and each user. While this feature has not been implemented, it could be incorporated into the existing system in the form of a resolver function (to be discussed in Section 3.4).

3. Another possibility is that the system has inferred (e.g. from user input or privacy rules) that Carol wants to release her location at a coarser granularity in certain contexts, including the current location query for Bob. This privacy management, again, can be supported by a spatial ontology that is able to blur the location granularity. The privacy aspect of the system is addressed by Carmichael (2008).

Personally Meaningful Location Labels

Carol’s current location is represented in a personally meaningful way (i.e. coffee room) to Alice. There are two levels of reasoning in presenting personally meaningful descriptions about location. One is to present a meaningful location description. So, for example, Alice may know of a local fruit shop, but does not know its address. So she would not understand it if a system reported that Bob was at that address. But she would understand it if the system stated that he was at Bondi Fresh Fruit. The other level is to personalise the location description. For example, the system may present the location as Room 125 to the general public who has never been there, and display it as coffee room to Alice because that is how she names that place. A personal ontology can be used to facilitate such reasoning by collecting information from various sources about people and location.

For the case of David’s office, which is Room 300, Bob sees a more personally meaningful place than the generic one given to Alice. The system infers that presenting David’s location as his office would be more useful to Bob. Such labels could be less useful at times; for someone who wants to physically find David and does not know where his office is, Room 300 may be a better choice for them to find it on a building map. On the other hand, if Bob knows where David’s office is, it is generally desirable
to bypass the room number, which people tend to be unaware of or forget. The system should be able to decide whether or not to show the more descriptive labels. The Adaptive Locator is capable of referring to a place by its owner, e.g. Bob’s Office, her office. For other descriptions of a place (e.g. coffee room), while the context ontology can model them, the Adaptive Locator has not been implemented to reason with that piece of information.

Selecting Relevant People by Inferred Social Networks

Eva’s current location is only relevant to Bob, but not to Alice. This may be, for example, inferred by the social network, such as Bob knows Eva, or based on other personal preferences, such as Alice may only want to know her colleagues’ locations during work hours. A personal ontology may be used to infer social networks by mining relevant sources, such as email communication. The Adaptive Locator displays people whom it infers to be known to the current user. A person is assessed as known to the user if they work at a nearby office or desk or if they are on the same internal mailing aliases.

Scrutability in Pervasive Computing Applications

One of the barriers for acceptance of adaptive systems is that they are prone to making mistakes in their presentation; these may include incorrect inferences, awkward user interfaces, use of out-dated information, and system faults. Thus, allowing users to scrutinise the system’s reasoning process (Bellotti and Edwards 2001) and to control the data about them (Rehman et al. 2005) are indispensable components in an adaptive system. Suppose that, based on the information provided from the example above, Bob goes to David’s office and cannot find him. Bob should be able to query the system, to see the reasoning process, and to determine that it infers David’s location from the location of his phone, which he accidentally left in his office. If David notices this false evidence, he then should be able to temporarily discard location evidence coming from his mobile phone. Previous work on user control and scrutability, which is also referred to as intelligibility, has mostly concentrated on user interface design (Bright et al. 2005; Czarkowski and Kay 2005; Tullio et al. 2007). The Adaptive Locator is capable of explaining system adaptation to the user, such as why the system infers that Alice knows David and why the system displays David’s location to Bob as “his office”.
1.2 Challenges

Accurate location information plays a critical role in context-aware systems. Modelling a user’s location is, however, difficult from both technical perspectives, such as precision and accuracy of the sensors, and non-technical aspects, such as the inconvenience of being required to carry a specialised sensor.

Depending on the application, it may be preferred to have location information presented at different levels of granularity: for example, when Bob’s manager wants to have a phone discussion with Bob, she may only want to know if he is at work or not: this may be inferred by his location, being either within the workplace building or not. If she knows that Bob is at work, she may prefer to call him using an internal line; otherwise, she might want to wait till he comes to work or choose a passive communicating means, such as leaving a note on his desk or sending email. By contrast, Bob’s daughter may want to know a finer-grain value for Bob’s location—Bob may be happy for her to know his precise location at all times—such as at the kitchen sink to have her homework book signed. To take just one other example, a Follow-me-music application may need to model Bob’s location accurately at the room level to deliver the desirable service.

Being able to present location information at different levels of granularity is an important aspect of location modelling in pervasive computing (Wishart et al. 2005). Determining accurate locations and tailoring the desired level of location granularity to users and contexts is challenging. This problem becomes more difficult in a multi-sensor environment, as more uncertainty may be introduced in resolving a location value from a range of location sensors (Hightower and Borriello 2004; Carmichael et al. 2005; Niu and Kay 2008a). Both the challenges of achieving accurate location modelling and the demands of presenting information at a context-relevant manner mean that it is important to design a system to model people’s location, to deliver personalised information, and to be able to explain the system’s adaptation.

This thesis addresses challenges in four aspects of a context ontology:

- Engineering a personal ontology for delivering contextualised and personalised information;
- Interpreting conflicting sensor evidence in a multi-sensor environment;
- Presenting personalised location labels with the personal ontology;
- Generating understandable explanations of the delivered personalisation.
CHAPTER 1. INTRODUCTION

The rest of this section discusses the challenges in each of those areas and how we tackle them.

**Engineering a Personal Ontology**

Ontologies are commonly used as static knowledge bases, for example, to more systematically associate tangible artefacts with abstract concepts (Hatala et al. 2005) and to establish common vocabularies for knowledge sharing (Chen et al. 2004a; Ranganathan et al. 2004b). Constructing a formal ontology is, however, a costly process in terms of efforts and time (Pinto and Martins 2004; Cimiano 2006). This is often referred to as the *knowledge acquisition bottleneck*. To overcome this problem, there has been much work on *ontology learning and population* in the early 90s, as reviewed in (Gómez-Pérez and Manzano-Macho 2003; Cimiano 2006). Although the ontology learning process is still far from fully automatic (Maedche and Staab 2001), there have been promising results in a number of projects. For example, the work by Maedche and Staab (2001);Navigli et al. (2003); Cimiano and Völker (2005); and other work reviewed in (Gómez-Pérez and Manzano-Macho 2003; Cimiano 2006) explore ways to semi-automate the process of ontology engineering.

This presents the first challenge in creating the MyPlace system: building ontological representations of each of the three layers of knowledge described on page 2. We have tackled this problem with different approaches for each of the three layers.

1. The first layer, being consistent for all buildings can be a carefully handcrafted ontology which captures the commonsense ideas that people have about all buildings, and about certain classes of buildings such as large workplaces.

2. The second layer, being specialised to a building should be cheaper to build. Essentially our approach has been to create a set of processes that can semi-automatically do this: for each new building, these processes need to be used to create the ontology that captures the important knowledge about that building.

3. The third layer is novel and challenging to build as it must capture knowledge and support reasoning that is dynamic over the lifetime of the building. Some of it is very dynamic as people move around the building, which relies on information from sensors that detect the location of people.
Interpreting Sensor Evidence

There are many sources of noise and uncertainty in reasoning about location from the evidence that can be collected by the many available sensor types. As a basis for the design of new ways that an ontology can address these difficulties, we identify three categories of problems in interpreting location evidence sources.

**Granularity variance.** This category describes the situation where there is a set of evidence, at different granularity levels. For example, a Bluetooth™ sensor, with a ten-metre range, may detect Bob’s Bluetooth-enabled phone, giving evidence of his location at the granularity of a wing of the building, such as Level 1 East. By contrast, a login sensor may provide evidence that he is at the finer grained location, at his desk in Room 100. If Room 100 is within Level 1 East, this does not represent conflict and a building ontology has the potential to reason that this is so. Similarly, if Room 100 is in Level 1 West and a finer-grain location than Level 1 is needed, there is a conflict and a building ontology has the potential to reason that this is the case.

**False positive.** This occurs when the available evidence is wrong. There are many ways this can happen. One important case occurs with sensors, such as Bluetooth detectors, which may detect Bob’s phone on two levels of a building. For example, one sensor on Level 3 may provide evidence and another on the floor above may also give evidence. A building ontology may be able to determine that these two pieces of evidence are in conflict. User behaviours may also cause false positive evidence: for example, Bob may leave his phone in Alice’s office, giving a stream of evidence that he is still in her office. If Bob then starts using his machine in his office, the activity sensor’s evidence of his location conflicts with the evidence from his phone location. A building ontology can be used to infer that this is a conflict.

**False negative.** This describes the case where evidence is missing. There are many potential causes: for example, a sensor may fail temporarily; there may be network latencies or failures. Importantly, people’s behaviours may play a role, for example, when Bob turns off his phone, or its battery fails. Another class of problem occurs when the Bluetooth sensor on Level 3 West stops delivering sensor readings and the sensors one floor above (Level 4 West) and below (Level 2 West) detect the presence of Bob’s mobile phone. We may infer that the Bluetooth sensor on Level 3 West has stopped working and that Bob is indeed located between Level 2 West
and Level 4 West (i.e. Level 3 West). This may require the ontology to specify the third dimension (i.e. the vertical dimension) of spatial relation.

Certainly, all of these can occur in combination. For example, a login sensor may provide evidence Bob is in Room 100, while two Bluetooth detectors may give evidence that his phone is both on Level 1 East and Level 2 East. The first two pieces of evidence may have granularity variance, the third is a false positive. An ontology of the building has the potential to support reasoning about the evidence, determining whether it is conflicting. In addition, ontological reasoning has the potential to help in the resolution of conflicting evidence, a key aspect that this thesis explores.

Ontological Reasoning for Personalisation

Referring again to Figure 1.1 on page 4, the location information of Carol and David are presented differently for different people: Coffee Room versus SIT Building and Room 300 versus his office. This illustrates one type of context-dependent information delivery: the same information is presented differently to different users. Another subtly different case is when the same information is presented differently even to the same user depending on the situation. So, for instance, David may want his location, being at the office, revealed only during work hours. When Bob queries David’s location during work hours, and David is working in his office, the location would be presented as his office; by contrast, if the query is submitted outside of work hours, the location might be presented as in Sydney, or simply unknown.

Consider the following scenario (with italics highlighting some key features):

Zack is an academic working in a multi-level building. Yvonne is an undergraduate student who comes to the building, wanting to discuss her assignment in Zack’s course. When she consults the Adaptive Locator interface at the foyer, seeking Zack, it tells her that Zack is on campus and will be available at his 2–3 pm official contact hour at Room 300. Later when Zack’s daughter Xena comes to his office, the Adaptive Locator tells her that Zack is in his office. Dubious Xena queries the Adaptive Locator regarding how her dad’s current location is inferred. An explanation of the resolution process is given, showing that there is conflicting evidence of Zack’s current location and a spatial ontology has been used to resolve it.

For a system to present Zack’s locations to Yvonne and Xena, there are several personalisation challenges to be addressed. To present Zack’s office as Room 300 to Yvonne
and as a more personalised form (i.e. *at his office*) to Xena, the system has to know the relationship between Xena and Zack as well as that Xena knows where Zack’s office is. Knowing that Zack has lenient privacy rules for family members, the system presents a fine grain of Zack’s current location to Xena (i.e. room-level granularity), by reasoning with a spatial ontology. By contrast, Yvonne is only presented with a coarser grained location: *on campus*.

**Explanations of System Adaptation**

The users should be able to scrutinise the information delivered to them (e.g. why does the system know I am in the tea room?) and have the control over information about themselves (e.g. I would rather not release my location information when I am away from the office). This requires the system to provide understandable explanations. In the scenario presented above, the system explains to the user about the identified conflicting evidence and its approach to resolve this conflict.

Figure 1.2 illustrates an example of such an explanation. It explains why and how the system presents Zack’s location as *his office* to his daughter. The explanation is roughly divided into two parts: location resolution and place description resolution. For the location resolution, it explains that there is ambiguous location evidence about Zack’s current location and Room 300 is chosen because it is spatially contained in Level 3 West, and the location value of Room 300 is given by the system sensor installed on Zack’s machine. If Xena wants, she may find out the list of ambiguous evidence sources by clicking on “click here to see recent sensor evidence”. For the place description

![Figure 1.2: An example explanation of location resolution by reasoning with a spatial ontology](image)
resolution, the system infers that Room 300 is Zack’s office and that Xena knows where
that is. If Xena would like more explanations about the reasoning process, she may
do so by clicking “why?”. Automating this explanation generation process requires a
suitable knowledge structure and reasoning mechanism.

1.3 Contributions and Thesis Outline

This thesis makes several contributions to pervasive computing, including resolution
of conflicting contextual evidence, personalisation, scrutability of personalisation, and
knowledge management. We briefly describe the main chapters and the contributions
each introduces.

Chapter 3 describes the Personalised Pervasive Scrutable Ontological Framework (PERSONAF), that consists of two main components: a central ontological knowledge
base—the Personalised Context Ontology (PECO)—and a reasoning component—
the Ontology- and Evidence-based Context Reasoner (ONCOR).

• Design of a conceptual framework to support ontological reasoning about location
  information in pervasive computing environments.

• Design of three ontological algorithms to reason about locations, including detection
  and resolution of conflicting sensor evidence.

Chapter 4 then presents a pragmatic approach to implement the PERSONAF frame-
work in an indoor pervasive computing environment. It also includes a preliminary
evaluation of PECO.

• Implementing the PERSONAF framework with an operational prototype.

• Designing a middle ontology to model the inside of a building by extracting concepts
  and the structure from a top-level ontology, OpenCyc (Lenat 1995); this facilitates
  pragmatic ontology population of light-weight ontologies, reusability, and
  interoperability.

• Demonstrating ways to learn and populate a context ontology with a range of
domain-specific sources.

• Creating reusable tools to build application ontology and accretion ontology adopt-
ing the PERSONAF framework.
Chapter 5 evaluates the upper two layers of PECO—the middle ontology and the application ontology—in terms of its ability to detect and resolve conflicting location evidence across different levels of location granularity.

- Demonstrating the power of an ontology to detect and resolve conflicting information in pervasive computing.
- Designing an experimental procedure for obtaining reliable records of people’s movements inside of a building.

Chapter 6 describes an adaptive location system back-ended by the PERSONAF framework and an qualitative evaluation of it. The user study evaluates the adaptive system in its ability to (1) deliver correct and useful personalisation, and (2) generate understandable explanations of the personalisation.

- Showing the capability of a personalised context ontology in personalising location information.
- Illustrating the power of a personalised context ontology in generating comprehensible explanations of the personalisation.

These chapters are preceded by a review of the critical background for the research, and the thesis concludes with a summary of the key contributions and future discussions for this research.
Chapter 2

Background

This thesis explores ways to exploit a personalised context ontology to support reasoning about indoor location information: in particular, location conflict resolution, personalisation, and scrutability. To provide background for this thesis, this chapter reports the state of the art of research, particularly drawing upon selected important work in three major areas relevant to the thesis: ontologies for pervasive computing, ontologies for personalisation and scrutability, and location conflict resolution. This chapter also introduces the location management framework, Personis/PersonisAM (Kay et al. 2002; Assad et al. 2007) that serves as the foundation of this thesis.

2.1 Ontologies for Pervasive Computing

Following Berners-Lee’s vision of semantic Web (Berners-Lee 2000), it is envisaged that information should be organised in a manner that is understandable and processable by machines (Kim 2002). A reasoning agent should then be able to efficiently deduce relevant information for each person in different contexts by reasoning with a large enough, semantically rich knowledge base, or ontology. This has great potential to support context modelling and personalised reasoning in pervasive computing. A large body of work has exploited ontologies to facilitate a number of important issues:

- interoperability: common understanding and knowledge sharing;
- contextualised and personalised reasoning and inference;
- context modelling;
- systematic validation.
Ontologies are difficult to evaluate. Even though there have been many attempts and suggestions reported in literature (e.g. work by Hartmann et al. (2004); Brank et al. (2005); Gangemi et al. (2006); Alani et al. (2006); Tartir and Arpinar (2007); Ye et al. (2007); Dellschaft and Staab (2008)), there is still no agreed upon evaluation framework, such as is available in information retrieval and data mining.

Brank, Grobelnik, and Mladenić (2005) usefully identified four common ways to evaluate an ontology: golden standard, application-based, data-driven and human assessment. Evaluation of an ontology can be further divided into six different dimensions: a lexical layer, hierarchy (i.e. is-a relation), other semantic relations, context/application layer, syntactic layer and structure and design. Depending on the structure and application of the ontology, different approaches and dimensions may be used.

Upon reviewing a representative work on the ontologies for pervasive computing, we identified four common approaches to evaluate a context ontology:

- application-based (Hatala and Wakkary 2005);
- performance-based (Wang et al. 2004a; Agostini et al. 2005; Lin et al. 2005; Forte et al. 2008);
- evaluation by implementation (Masuoka et al. 2003; McGrath et al. 2003; Ranganathan and Campbell 2003a; Chen et al. 2004e; Christopoulos et al. 2005; Heckmann et al. 2005; Kim et al. 2006);
- scenario-based (Chen et al. 2004a; Gu et al. 2004; Agostini et al. 2005; Christopoulos and Kameas 2005; Strimpakou et al. 2006; Ejigu et al. 2007).

The last two approaches appear to be widely used, and this is because of the nature of pervasive computing, where applications may differ significantly across different domains, and practicality is often the central concern.

The rest of this section reviews representative work that illustrates approaches that explore ways to make use of ontologies to address various aspects of context modelling and reasoning in pervasive computing. For each context ontology, we include the following components:

- motivations for constructing and using the ontology;
- its design and structure;
- the reasoning mechanism;
• the evaluation of the ontology;
• a summary and the key contributions of that ontology or the system which adopts it.

Ontologies in GAIA

Ranganathan and Campbell (2003b) is pioneering work towards the use of ontologies to describe context in the pervasive computing environment. The ontologies are integrated into the context-aware infrastructure, GAIA to achieve the following goals (McGrath et al. 2003; Ranganathan and Campbell 2003b; Ranganathan et al. 2004a):

• establishing common understanding between agents by defining terms used in the pervasive computing environment;
• systematically validating ontological descriptions of entities and contextual information;
• matchmaking by discovering relevant and compatible entities, whose meta-data are represented in ontologies;
• context reasoning with logic.

The ontologies used in GAIA are managed by a centralised Ontology Server, which serves logical queries and responds with relevant contextual information about those ontologies. Figure 2.1 shows a partial view of the hierarchical relationships between the entities in GAIA. The ontologies represent meta-data about each class of entity, and each class instance also has its properties described in an ontological form.

GAIA allows a variety of reasoning mechanisms to be used with querying higher-level contextual information, reasoning about application behaviour, and reasoning about contexts (Ranganathan and Campbell 2003b; Ranganathan et al. 2004a). It provides two main streams of reasoning mechanisms: logic and machine learning. In particular, Ranganathan et al. (2004a) discusses in detail how GAIA supports applications and services to reason about uncertainties with probabilistic logic, fuzzy logic, and Bayesian networks.

This work is important as one of the first attempts to integrate semantic representation and reasoning into an implemented pervasive computing system. It has been shown that it is feasible to integrate ontologies into the prototype infrastructure GAIA, which in turns provides potential for modelling a wide variety of entities in the pervasive computing environment (McGrath et al. 2003; Ranganathan and Campbell 2003b;
Figure 2.1: Part of the entity hierarchy of GAIA, reproduced according to the image in (Page 26 of McGrath et al. 2003)

Ranganathan et al. 2004a). This work was validated by implementing the framework into a real pervasive computing application.

**Context Broker Architecture Ontology (COBRA-ONT)**

Context Broker Architecture Ontology (COBRA-ONT) (Chen et al. 2003, 2004a) is a set of ontologies constructed for the Context Broker Architecture (CoBrA) (Chen et al. 2004b,c), which is a context management architecture for the pervasive computing environment. The goal is to address two shortcomings of previous systems: knowledge sharing and context reasoning. Essentially, the system tries to determine the location and the activity of an agent (i.e. human or software). A prototype, EasyMeeting back-ended by COBRA-ONT is reported. EasyMeeting models the user and the context to provide context-aware services in a smart meeting room. An informal evaluation was reported; this was based on the use and feedback from three groups of people, mainly guests, who tried the system.

Three different versions of COBRA-ONT have been reported in literature. COBRA-ONT v0.2 (Chen et al. 2004a) was expressed in OWL-DL and it aimed to model places, agents, and activities. COBRA-ONT is categorised into four themes: physical places, agents (humans or software agents), location of the agents, and activity of the agents. A place (e.g. a lab, restroom) is either an AtomicPlace (i.e. smallest place unit) or a CompoundPlace which are both subclasses of Place. An agent is associated with one or more roles (e.g. speaker and audience) in each event.
The next version of COBRA-ONT (v0.3) (Chen et al. 2004b) includes a location ontology, a device ontology, and a temporal ontology. The physical location ontology includes geographic boundaries (e.g. rooms, buildings), spatial properties (e.g. atomic places, compound places), and temporal properties (e.g. meeting rooms during the working hours, offices on a public holiday). Based on spatial properties, it is possible to develop a hierarchy between locations. Then in a latter version of COBRA-ONT (Chen et al. 2004c) (shown in Figure 2.2), it is expanded with four more sets of ontologies: actions, agent, meeting, and document.

Figure 2.2: The structure layout for COBRA-ONT v0.4, reproduced according to the image in (Page 180 of Chen et al. 2004c)
Device context may include basic knowledge about the device (e.g. is it Bluetooth enabled), the device-ownership relationship (e.g. who is the owner(s) of the device?), temporal properties (e.g. when was the last time a device was used) and spatial properties associated with a device (e.g. what is the distance between a device and its owner). They adopt a part of the ontology from the IEEE FIPA device ontology specification\(^1\), which is an ontology that describes technical details of a device (e.g. software and hardware specifications). The temporal ontology that adopts the DAML-Time ontology provides a way to describe an instant and an interval of time (e.g. a day, an hour), but it does not describe temporal relations between events as SOUPA does, to be described next.

A rule-driven logic inference engine F-OWL (Zou et al. 2004) was developed to support ontological reasoning in COBRA-ONT. A scenario-based validation was reported about this work.

**Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA)**

Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA), an OWL based ontology, serves as shared upper/middle ontologies for ubiquitous computing. Chen et al. (2004e) and Chen et al. (2005) present two parts of SOUPA: SOUPA Core and SOUPA Extension; the latter was developed mainly to demonstrate how the SOUPA Core ontologies can be extended. The SOUPA Core consists of nine ontologies: person, policy, action, time, geo-m, event, space, BDI (believes, desire, intention) and agent. Many of the vocabularies were borrowed, rather than imported, from existing ontologies: FOAF for person, DAML-Time for time, spatial ontologies in OpenCyc and Regional Connection Calculus (RCC) for space, COBRA-ONT, MoGATU BDI ontology, and the Rei policy ontology. The time and person ontologies use the standard OWL ontology mapping constructs (i.e. `owl:equivalentClass` and `owl:equivalentProperty`) to map the borrowed terms to the top-level ontology.

The person ontology includes basic information, contact information, and a social and professional profile. An agent may be a person or a computational entity and is characterised using the BDI ontology. In addition to describing a set of *mentalistic* notions, such as knowledge, belief, intention, and obligation, BDI also describes desire, conflicts and feasibility.

A policy is defined as 'a set of rules that is specified by a user or a computing entity to restrict or guide the execution of actions'. In other words, it defines *who* may access *what* services. A policy is often context-dependent: *who, under what conditions,* may access what services, and that context component seems to be omitted in the design.

For the ontology of time, it adds, on top of the temporal ontology of COBRA-ONT, temporal relations (e.g. X starts before Y) of events, which may be important during reasoning process.

A feasibility evaluation is reported by presenting use case scenarios on two applications: CoBrA (Chen et al. 2004c), which supports context-aware services in a smart meeting room, and MoGATU (Perich et al. 2004), which supports peer-to-peer semantic data management in a pervasive computing environment by modelling the user’s beliefs, desires, and intentions.

This work is important as an early and ambitious exploration of reasoning about context, making use of ontologies and other inference techniques. It is also important as an exploration of the ways to support privacy in pervasive computing Applications, by making use of such sophisticated reasoning about people, time, locations and other contexts. Notably, it is difficult to evaluate such systems and the authors have chosen to do this by incorporating it in an illustrative prototype which was subject to qualitative evaluation.

CONON

CONON (Context Ontology) (Gu et al. 2004; Wang et al. 2004b) is a context ontology represented in OWL (OWL-Lite) for modelling contexts in pervasive computing environments. The aim is to build an upper ontology to support context modelling and logic-based reasoning, as well as to facilitate extension for more domain- and application-specific ontologies. In (Gu et al. 2004; Wang et al. 2004b), the Upper-level Context Ontology (ULCO) contains 14 concepts (see Figure 2.3), including four most fundamental concepts: location, person, activity and computational entity. Wang et al. (2004a) extends the ULCO with one more level of location concepts, such as room, building, and city. Gu et al. (2005) further refined the upper ontology by both inserting an addition fundamental concept of time and enriching it with inter-relations between the five fundamental entities, such as Person is located-in Location.

Two types of reasoning are used in the scenarios described in (Gu et al. 2004; Wang et al. 2004b): ontological reasoning using description logic (e.g subsumption, transitivity) and user-defined forward-chaining rules using first-order logic (e.g. dimmed light, low level of noise, and a person lying on the bed may indicate that the person is sleeping). The Jena2 Semantic Web Toolkit (Carroll et al. 2004) is used to build logic reasoners for both methods. In addition to forward-chaining reasoning, Gu et al. (2005) also use backward-chaining and a hybrid execution model to perform user-defined rule-based reasoning.
Each context (e.g. Bob is located-in Room 300) is classified into one of the four categories (Gu et al. 2004): sensed context, defined context, aggregated context and deduced context. The former two are further grouped into direct contexts and the latter two are indirect contexts. Each type of context may give different confidence levels in the reasoning process. The confidence level is determined by the quality of the contextual information, which is quantified based on accuracy, resolution, certainty and freshness. For example, the quality information for a location context could be the following: it was given two minutes ago in geographical coordinates with a resolution of 10 metres and an accuracy of 80%.

Gu, Wang, and their colleagues conducted evaluations based on a number of performance studies on real-time context reasoning (Wang et al. 2004b,a; Gu et al. 2005). The performance studies were based on response times for context querying and reasoning.
on a range of data sizes, from 1,000 to 10,000 RDF triples. Gu et al. (2005) additionally evaluated the ontology based on loading and merging the two layers of ontologies and context discovery. The response time was roughly proportionally to the size of the context data, with ontological reasoning being more computational demanding than logic-based rules reasoning. The results demonstrated “reasonable performance” (page 16 of Gu et al. 2005) and showed potential for applying this ontological approach in the pervasive computing environments for non-time-critical applications.

This work proposes a promising upper-level ontology to model the pervasive computing environment and is important for being one of the first performance-based evaluations of ontological reasoning in pervasive computing.

**UbisWorld Ontology**

Another important example of a middle/upper context ontology for pervasive computing is the UbisWorld Ontology, or UbisOntology (Heckmann 2006). It is a collection of concepts and models that aims to facilitate context modelling, ontological representation, and knowledge sharing for pervasive computing and user modelling. UbisOntology consists of six additive ontologies, namely the physical ontology, the spatial ontology, the temporal ontology, the activity ontology, the situation ontology, and the inference ontology.

Motivated by the need to exchange heterogeneous user models between different user-adaptive systems, the General User Model Ontology (GUMO) (Heckmann et al. 2005) has been constructed as part of the situation ontology in UbisWorld. GUMO models typical dimensions in user models, and many of these are based on the Suggested Upper Merged Ontology (SUMO) (Niles and Pease 2001), the Mid-Level Ontology (MILO) (Niles and Terry 2004), and a number of psychological theories, such as the Three Factor PEN Model (Oberlander and Gill 2004), Five Factor OCEAN Model (Oberlander and Gill 2004), and MyersBriggs Type Inventory (Matthews et al. 2003).

UbisWorld is defined in OWL (OWL-full) and uses **SituationalStatements** (Heckmann 2003) to represent its statements about all user model aspects. A **SituationalStatements** is in the following format:

```
subject { auxiliary, predicate, range } object
```

An example would be **William { hasInterest, contract bridge, low-medium-high } High**, which, in plain English, means “William has high interest in contract bridge”. In addition to the **SituationalStatements**, the U2M (ubiquitous user modelling) namespace is
used to enrich the GUMO and the UbisOntology with a list of useful metadata attributes, such as an identifier, the expected expiration time, and relevant images of the concept.

For location modelling, it uses both a spatial topology to derive a location path (see Figure 2.4(b)), as well as the UbisOntology (see Figure 2.4(a)), that describes the UbisWorld. For example, Room 300 in the SIT building may be represented as in the UbisWorld as

UbisWorld -> Spatial and Temporal Elements -> Spatial Element -> Location -> Room -> Room 300,

as well as in the actual spatial topology as

Any Place -> Oceania -> Australia -> New South Wales -> University of Sydney -> SIT Building -> 3rd Floor -> Room 300.

The ontologies have been incorporated in a number of projects, including speech recognition, physiological data sensing, shopping assistance, museum tour guide, and pedestrian navigation (Heckmann 2006). Those applications indirectly evaluate the validity of the UbisOntology and strongly suggest its usefulness in modelling users and other entities in pervasive computing with Semantic Web technologies.

This work is important for its high practicality—implied from the wide variety of

![Spatial Topology](image1.png) ![Spatial Ontology](image2.png)

Figure 2.4: Spatial Representation in UbisWorld in the Ontology Browser on ubisworld.org (accessed 23 August 2006)
applications adopting it—and its large-scale ontological representation of the blocks world that simulate, monitor, and control the pervasive computing environment. The UbisWorld has also been made available on the Web since, at latest, March 2005\(^2\), as a user modelling server.

Gadgetware Architectural Style (GAS) Ontology

The Gadgetware Architectural Style (GAS) is an architectural framework that supports pervasive computing systems in ways that embed computational power into everyday artefacts (Kameas et al. 2003). The GAS Ontology (Christopoulou and Kameas 2005) is further introduced to enable intelligent interaction between the artefacts, or eGadgets. The focus is, therefore, to establish common understanding between the eGadgets inside the eGadgetWorld, so as to facilitate knowledge sharing and service discovery.

The GAS ontology is designed to model the eGadgetWorld with two layers: the GAS Core Ontology (GAS-CO), which is a static common language for all eGadgets to communicate with one another, and the GAS Higher Ontology (GAS-HO), which represents private knowledge for each eGadget, including static information of that artefact (i.e. GAS-HO-static) or dynamically acquired contextual knowledge (i.e. GAS-HO-volatile). The first version of GAS-CO (Christopoulou and Kameas 2005) consists of seven classes that provide a complete representation of an eGadget: eGadgetWorld, eGadget, Plug, TPlug, SPlug, Synapse, and Service. The next version of GAS-CO (Christopoulou et al. 2005), as shown in Figure 2.5, has two additional classes: Parameter and State, that enable more specific state activation rules; for example, when noise level is less than 30 decibels, change the state of the lamp to 10% brightness.

The reasoning power is provided by the GAS Ontology manager. Its main responsibilities are to facilitate smooth communication between eGadgets and to provide service discovery. As each eGadget may be connected to another eGadget through a synapse with compatible plugs, the service that the newly formed eGadget provides can be different from the individual eGadgets. The Ontology manager then provides compatibility checks between plugs and discovers the updated set of services the newly formed eGadget provides.

The GAS Ontology is evaluated through careful analysis of a scenario (Christopoulou and Kameas 2004, 2005) and analysis of another scenario with an implemented prototype system using six eGadgets (Christopoulou et al. 2005). The results show potential for this context management and reasoning framework.

This work is important as it is one of the first systems that provides service description and discovery with distributed ontologies in a pervasive computing environment.

**Ontologies in ec(h)o**

The ec(h)o (Wakkary et al. 2003) is a recommender system integrating audio, vision, and location tracking. The aim is to provide a personalised museum tour augmented with three-dimensional soundscapes. This is done by accounting for the user’s interests—from explicit user selections of sound objects—and physical movements throughout the visit.

To address the shortcomings from the previous systems that either the content was fixed or the content access was limited locally to one museum, the Semantic Web technologies are adopted to facilitate networked, dynamic, and user-driven content delivery (Hatala et al. 2005). Ontologies were, thus, used to formally describe the museum artefacts, annotated sound objects, and the psychoacoustics and sound characteristics of the sounds objects (Hatala et al. 2005). As the ontologies can provide a formalism to connect the digital content, user environment, and user interests, they make knowledge sharing across multiple applications possible (Hatala and Wakkary 2005).

The ontologies are primarily built on the standard Conceptual Reference Model (CRM) (Crofts et al. 2006) for heritage content. Figure 2.6 is a partial view of how each ontology in ec(h)o inter-relates to the others. They facilitate interoperability between
other museums by mapping the topics to the Dewey Decimal Classification\(^3\) whenever possible.

There were a total of 56 classes (225, including the upper ontologies) with 1033 instances (2433, including short sound objects introducing the main objects), resulting in 65505 RDF triples during the reasoning process (Hatala et al. 2005). The reasoning was done with user-defined forward-chaining rules.

The system went through extensive testing and evaluation, both laboratory and on-site, before the actual deployment for an exhibition at the Nature Museum in Ottawa in March 2004. Although a separate evaluation of the ontologies is not reported in literature, the results for the system evaluation indicated that the ontologies and rule-based approach is a promising combination in delivering highly-responsive context aware information. However, the authors expressed that it was hard for the users to follow how subsequent sound objects relate to each other, because of the chosen level of abstraction to inter-connect the sound objects; the sound objects were linked with each other in a higher level of concepts than merely the story-line (Hatala and Wakkary 2005).

This work is important as it provided one of the first user evaluations with real deployment of a pervasive computing system that integrates an ontological approach to

represent the entities and the contextual relationships.

Summary

We have presented six representative projects that make use of context ontologies in pervasive computing. The pioneering work by Chen et al. (2003) and Ranganathan and Campbell (2003b) laid a foundation for later work (e.g. Strang et al. (2003b); Becker and Nicklas (2004); Agostini et al. (2005); Lin et al. (2005); Strimpakou et al. (2006); Kim et al. (2006); Ejigu et al. (2007); Forte et al. (2008); Hu et al. (2008)) in the direction of semantic modelling and ontological reasoning in the pervasive computing environment. Wang et al. (2004b); Gu et al. (2005) made a valuable contribution to this area of research by presenting one of the first performance-based evaluations in terms of real-time context reasoning on their work. Heckmann (2006) tackles the important problem of ubiquitous user modelling by presenting a comprehensive upper ontology for user modelling (Heckmann et al. 2005) and an enduring ubiquitous user modelling server on the World Wide Web. Christopoulou and Kameas (2005) proposes a promising pervasive computing infrastructure that provides formal and interoperable service description and discovery with distributed context ontologies. Hatala and Wakkary (2005) presents one of the first user studies on a fully implemented and deployed pervasive computing system in a real museum exhibition, which integrates an ontological approach to semantic representation of the entities.

As the focus of this thesis is to demonstrate the use of the PERSONAF framework, we use fundamental ontological algorithms for location resolution. However, the flexibility of this framework allows easy adaptation of additional resolver functions, such as prior probability and Bayes filters (Hightower and Borriello 2004).

In terms of ontologies, the reviewed systems still need better separation between upper (information exchange) and lower (information collection) layers (Ye et al. 2007). The ontological framework introduced by this thesis makes this distinction more clear, basing it on a handcrafted location ontology (Kay et al. 2007) from an existing ontology, OpenCyc (Lenat 1995). We also demonstrate the power of this framework by implementing and evaluating it in a working demonstrator system.

2.2 Personal Ontology and Personalised Location Labels

While Section 2.1 gives an overview of ontologies used in pervasive computing, this section focuses on two other areas related to this thesis: ontologies that facilitate user modelling and approaches to generate personalised location labels.
Personal Ontology

The notion of personal ontology departs from mainstream ontological research, which seeks common understandings that can be represented in ontological models to be shared by many applications. However, there has been some work that has aimed to capture the individual user’s understanding of a domain (Tennison and Shadbolt 1998; Dieng and Hug 1998; Huhns and Stephens 1999; Kim et al. 2001). A somewhat different form of personal ontology aimed to model a user’s interests in one or more domains (Katifori et al. 2005). Yet another form of personal ontology aimed to create an ontological representation of a user model or user profile (Gauch et al. 2003; Middleton et al. 2004; Golemati et al. 2007; Vildjiounaite et al. 2007; Katifori et al. 2008).

Previous work has explored various issues associated with personal ontologies. For example, there has been work on comparison and fusion in personal ontologies (Dieng and Hug 1998); document searching and browsing (Huhns and Stephens 1999; Kim et al. 2001); context-awareness (Vildjiounaite and Kallio 2007; Vildjiounaite et al. 2007); Web navigation (Chaffee and Gauch 2000; Gauch et al. 2003; Noh et al. 2003; Trajkova and Gauch 2004; Gauch et al. 2007); personal ontology elicitation (Tennison and Shadbolt 1998; Tennison et al. 2002; Cimolino et al. 2004; Katifori et al. 2008); and in personal information management (Katifori et al. 2005; Golemati et al. 2007; Dix et al. 2007).

In this large body of work, the personal ontologies were generally managed in a user-centred, distributed manner: each user would have an ontological profile of contextual information, similar to a hierarchical user model, and that is normally limited to modelling users. By contrast, we introduce a more modular architecture: we have ontologies for users, location and devices, and each proposition in the ontologies has at least one source that may give evidence relevant to individuals and/or groups.

Personalised Location Labels

Hightower (2003) introduces the problem of moving from location to place, which points out the need to present a position in a human-readable form. A formative user study reported by Consolvo et al. (2005) further suggested that even symbolic names like street addresses are seldomly a user’s first choice when they disclose their location to other people. This thesis, therefore, aims to explore ways to present location in a more personally meaningful form to a user by reasoning with a personal ontology.

As identified by Zhou et al. (2007), there are two essential steps in personal place acquisition: obtaining physical locations and obtaining labels. There is a large body of work in the area of automatically obtaining physical locations, for example (Ashbrook
2.2. PERSONAL ONTOLOGY AND PERSONALISED LOCATION LABELS

and Starner 2003; Hightower et al. 2005; Kang et al. 2005; Zhou et al. 2007; Nurmi and Bhattacharyya 2008), and this is not the focus of this thesis. Instead, this thesis focuses on the second step of personal place acquisition: obtaining labels, more specifically, personalised location labels.

Zhou et al. (2007) points out the common ways to obtain a location label:

- User-defined place labels. Those are labels explicitly given by users themselves; one example of this approach is presented by Smith et al. (2005) in their user study to obtain a list of places. They are typically the more meaningful labels to the users, but labelling a large number of places can be time-consuming.

- Community-based collaboratively defined place labels. A user may choose to share their place labels with other users within a community. This approach may reduce labelling workload at the user-side, and the amount of workload reduction depends on the number of users sharing their labels and the number of labels shared. The possible down side is that some labels may be highly personalised, such as “Mom’s favourite coffee shop”, and, therefore, is of little value to other users.

- Generating place labels from geo-databases and gazetteers. Reverse geocoding (Krumm 2007) can produce a postal address from geographic coordinates, i.e. a latitude/longitude pair. Naaman et al. (2004) further developed an algorithm to infer symbolic names, such as “Stanford” and “Around San Francisco”, from geo-databases. This is a promising approach to quickly obtaining meaningful place labels, but it would not be personalised. Moreover, depending on the algorithm used to learn the place labels, they might suffer from inaccurate inferences.

- Application-based place labels. There is a range of applications (e.g. Google Maps, Yahoo! Fire Eagle, Twitter, CellSpotting.com) that allow or potentially allow users to label various locations, such as Central Park, dad’s office. This approach is similar to the community-based labels, but can potentially reach a larger number of users.

This thesis proposes another way of obtaining personally meaningful location labels, by way of ontologically reasoning with a personal ontology. There has been a rather limited exploration of ways to obtain personalised location labels by reasoning with ontologies. Alves et al. (2007) proposes a two-stage approach to natural language generation of location description: (1) enriching available locations with a Generic Place
Model Ontology and (2) generating textual descriptions of the locations from the resulting enriched representation of the location. The detailed description of that place ontology and its evaluation are, however, not available in the literature.

2.3 Location Conflict Resolution

One critical property for most context-aware systems is being able to resolve conflicting context. Many contributions have been made in this area. The inherent challenge in conflict resolution, or ambiguity reduction, is the heterogeneity of context observed in a pervasive computing environment. This often requires either a mediation approach involving users (Dey et al. 2000) or a rather ad-hoc, rule-based approach (Ranganathan and Campbell 2003a; Aly 2007) to address the specific conflicting context.

On the other hand, approaches that involve the use of machine learning techniques (Castro et al. 2001; Hightower and Borriello 2004; Ranganathan et al. 2004a) are more systematic and, therefore, more practical in the pervasive computing environment with potentially large quantities of heterogeneous context information. The problems those learning algorithms often face are the cold-start problem and the inscrutability, or unintelligibility (Bellotti and Edwards 2001) in terms of the inability to show the reasoning process.

We tackle this problem by using an ontology to reason about conflicting location information. The rich semantics and explicit relationships of an ontology makes it a good candidate to provide systematic and scrutable reasoning about conflicting context information. This process involves two key steps: conflict detection and conflict resolution.

Myllymaki and Edlund (2002) presents a location resolution algorithm using the evidence of multiple mobile objects. Wishart et al. (2005) adopts a similar approach to Myllymaki and Edlund (2002); their work uses two additional confidence factors: discounting the weights of the sensor sources that previously provided incorrect readings and accounting for activeness of each sensor type. Both include a step to address the granularity variance conflict, but the actual underlying approach is not documented.

Chen (2004) proposes a similar ontological approach to address granularity variance about location and uses assumption-based rules to decide the more reliable source by invalidating false assumptions. Wishart, Henricksen, and Indulska (2007) proposes an obfuscation approach to address the issue of partial location privacy, which can “adjust the granularity of different types of context information to meet disclosure requirements stated by the owner of the context information.” The literature, however, did not report
an implementation of an actual location ontology in a working prototype.

In a more general sense of context conflict resolution, Ranganathan and Campbell (2003a) addresses the problem with *priority-based rules*. This is, more weight is assigned to the entities (e.g. tasks, events, objects) that take higher priority. If the priorities are the same, the system randomly chooses one of the actions to execute.

## 2.4 Personis and PersonisAM

We now introduce the location management framework, Personis (Kay et al. 2002) that serves as the foundation of this thesis. The main idea of Personis is its *accretion and resolution* (A/R) representation. It *accretes* evidence and *resolves* a value that is both time and context relevant in real time. We call a function that resolves a value with the accreted evidence a *resolver*. A/R has been designed for simplicity, flexibility and, critically, for user control and scrutability. There are several advantages of using A/R, especially in a pervasive computing environment:

- Processing and reasoning is delayed until a result is actually needed, saving unnecessary computation and providing more flexibility.

- Scrutability can be supported by making information about the functions and data used available in explanations about the reasoning for the user.

- The use of different resolver functions allows different results to be returned depending on the context and identity of the person or process asking for the value.

- The use of resolver for delayed value interpretation facilitates dynamic adoption of extra reasoning mechanisms, such as a context ontology (Niu and Kay 2008b) or a Bayes filter (Hightower and Borriello 2004).

We now present an example to illustrate this approach. Table 2.1 shows a sample of three pieces of location sensor evidence (top table) and how they would be resolved into different values by applying three different resolvers on them (bottom table).

If we assume that this is the evidence available about a user’s location at 18:12, 22 July 2008, we can see that the newest evidence is 2 minutes old and indicates that user is at Level 1 East. It comes from a Bluetooth sensor which detected the user’s phone. Four minutes earlier, the user’s system activity sensor provided evidence that the user was at Desk 3W32, and ten minutes earlier the user was detected at Level 1 West by another Bluetooth sensor. The accretion step involves collection of evidence such as that
Table 2.1: An sample list of location sensor evidence (top) and resolved values from three different resolvers (bottom)

<table>
<thead>
<tr>
<th>Detection Time</th>
<th>Location</th>
<th>Sensor Type</th>
<th>Detected Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:10, 22 July 2008</td>
<td>Level 1 East</td>
<td>Bluetooth sensor</td>
<td>Bob’s Mobile Phone</td>
</tr>
<tr>
<td>18:06, 22 July 2008</td>
<td>Desk 3W32</td>
<td>System activity sensor</td>
<td>Bob’s Office PC</td>
</tr>
<tr>
<td>17:56, 22 July 2008</td>
<td>Level 1 West</td>
<td>Bluetooth sensor</td>
<td>Bob’s Mobile Phone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolver</th>
<th>Resolved Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>Level 1 East</td>
<td>The most recent evidence</td>
</tr>
<tr>
<td>Ex-Bluetooth</td>
<td>Desk 3W32</td>
<td>Excluding evidence from Bluetooth sensors</td>
</tr>
<tr>
<td>Ex-after-hour</td>
<td>Level 1 West</td>
<td>Excluding evidence generated after work-hours</td>
</tr>
</tbody>
</table>

in the table. The three simple resolvers are now described to illustrate the resolution step:

**Point.** This algorithm simply fetches the most recent piece of evidence from the available ones (Hightower and Borriello 2004). From the available list of evidence, it would return the location value of Level 1 East, as it is the evidence with the most recent timestamp.

**Ex-Bluetooth.** As one of the resolvers used by Carmichael (2008) for privacy control, it filters out evidence coming from Bluetooth sensors. Therefore, it would return the value of Desk 3W32, the only evidence that was not from a Bluetooth sensor.

**Ex-after-hour.** This is another resolver used by Carmichael (2008), which filters out evidence generated outside of work-hours (i.e. 9am–5pm weekdays). This then returns the value of Level 1 West, which is the only evidence generated within work-hours.

Note that A/R requires details of the evidence source for each piece of evidence. For example, the source of the first piece of evidence in Table 2.1 is a Bluetooth sensor.

This A/R approach saves unnecessary computation at the time of evidence accretion and allows adaptive interpretation as well as scrutiny of the accumulated evidence. The trade-off is an increase in real-time computation; this, however, has been shown to be manageable (Carmichael, Kay, and Kummerfeld 2005). Carmichael, Kay, and Kummerfeld (2005) describes an A/R approach to the generalised modelling of the pervasive computing environment, specifically users, devices, and places. A large-scale performance-based evaluation demonstrated the scalability of this framework. Carmichael (2008)
2.5. SUMMARY

further explores its capability to manage privacy in the pervasive computing environment through a user study with an operational prototype.

Assad et al. (2007) extends the Personis framework with active and distributed modelling. An active model is one with triggers associated with it, which ensures timely updating of the model as results of updates of depended models. Being able to ensure consistent and generalised modelling is crucial in the highly dynamic environment of pervasive computing.

The other main property that A/R is designed for is scrutability. Allowing users to scrutinise the system’s reasoning process (Bellotti and Edwards 2001) and to control the data about them (Rehman, Stajano, and Coulouris 2005) are critical in an adaptive pervasive computing system. This aspect of A/R approach has been extensively evaluated by Czarkowski, Kay, and Potts (2005), Kay (1995), and Lum (2007).

We chose to build our framework upon the A/R approach with PersonisAM for its consistent and generalised modelling of the pervasive computing environment (Carmichael et al. 2005; Assad et al. 2007), its support for scrutability which has gone through extensive evaluation, and the availability of infrastructure based on it (Assad et al. 2007).

2.5 Summary

This chapter provides background for the thesis, reviewing related work about the potential of an ontology to model the pervasive computing environment, to reason about users’ understanding, interests, and other attributes about the user, to generate personalised location labels, and to resolve conflicting context information. In addition, we introduced PersonisAM, the location management framework this thesis builds upon.
Chapter 3

The PERSONAF Conceptual Framework

Ontologies have great appeal in systematically modelling domain knowledge, but they have largely been used as static knowledge bases, for example, to formally describe tangible and abstract concepts (Ranganathan and Campbell 2003b; Hatala et al. 2005) and to establish common vocabularies for knowledge sharing (Strang et al. 2003b; Chen et al. 2004a; Ranganathan et al. 2004b). For modelling the pervasive environment, this may constrain the flexibility in personalised and contextualised reasoning (Sowa 2006).

To organise information of the magnitude of the World Wide Web, it is critical to somehow automate the information processing. To tackle this so-called information bottleneck problem, researchers started to semi-automate the process to construct, update, and populate an ontology, which is called ontology learning and population (Maedche and Staab 2001; Cimiano 2006).

Inspired by automatic ontology learning and population, we apply this technique to readily available document sources to model the highly dynamic pervasive computing environment (Kay, Niu, and Carmichael 2007). To better capture the dynamic interaction of a pervasive computing environment, we use a dynamic context ontology with the accretion and resolution (A/R) approach. We call it the Personalised and Scrutable Ontological Reasoning (PERSONAF) framework.

Figure 3.1 illustrates the main elements in the PERSONAF framework. Central to this conceptualisation are these elements. At the upper left, we show sources of external information. The figure shows that one class of these comes from building plans, which have the potential to provide evidence for ontological reasoning about the building. The

---

1 A shorter version of this chapter have been published in the workshop proceedings UMI 2008.
other important class, shown in the lower set of sources, provides evidence to building personal ontologies about people. So, for example, an email list may be used to infer relationship between all the people on that list in a workplace, such lists may define important groupings of people.

There are two main components in this framework, highlighted with bold text and boxes with thicker borders: a central ontological knowledge base—the Personalised Context Ontology (PECO)—and a reasoning component—the Ontology- and Evidence-based Context Reasoner (ONCOR). At the upper right, we show the top-level ontology, a standard, existing ontology, OpenCyc. Below this are the three layers of PECO: the Middle Building Ontology, MIBO, that captures the ontological model for a generic building and this may be integrated with or linked to the top-level ontology; an Application Ontology which can be constructed from a set of building maps to model a particular
building; and the personalised and light-weight layer of the *Accretion Ontology*. The accretion ontology accumulates propositions mined from various sources and stores them as evidence. PECO holds all the ontological information and then, when an application needs to access the information, the *Value Resolution* process is performed by ONCOR: this has a collection of alternative processes for interpreting evidence within PECO, as shown in the lower box of Figure 3.1. On the left of the ONCOR box in the figure, there is a collection of *resolvers* that have been specifically created for reasoning about location—the central context for this framework—and those use ontological reasoning. We now describe each layer of PECO and how ONCOR interprets the PECO.

### 3.1 Middle Ontology

Reusing a top-level ontology in engineering a domain or task ontology has potential to save time and resources, especially when the resulting ontology needs to be interoperable with other applications that are able to interpret the same ontology. A number of top-level ontologies have been proposed for this purpose: for example, WordNet (Miller et al. 1990), OpenCyc (Lenat 1995), Sowa’s ontology (Sowa 1995), SUMO (Niles and Pease 2001), and DOLCE (Gangemi et al. 2002).

From a pragmatic point of view, a top-level ontology is often too general and too computationally expensive for modelling and reasoning about a relatively small set of concepts. As a result, a more specialised and domain-specific ontology is often more suitable, especially for real-time reasoning. This ontology can consist of a relatively small set of concepts and relationships that can be crafted from the top-level ontology. We call this a *middle ontology*\(^2\). As reviewed in Section 2.1 on page 14, there are several proposals for such ontologies across different domains in pervasive computing: for example, SOUPA (Chen et al. 2004e) and CONON (Wang et al. 2004b) focus on context modelling in pervasive computing environments; and GUMO emphasises on user modelling for the Semantic Web (Heckmann et al. 2005).

A middle ontology is usually extracted from a more generic, top-level ontology, possibly with minor alterations. The process typically includes the following steps, similar to the ontology engineering methodology described by Noy and McGuinness (2000):

1. Choose a domain or domains that requires formal specifications for machine reasoning, e.g. space, devices.

---

\(^2\) This term is borrowed from the Mid-level Ontology (MILO) (Niles and Terry 2004), which is a bridging ontology for SUMO and various domain ontologies under SUMO.
2. Choose a top-level ontology that provides suitable specifications about the domain(s), e.g. OpenCyc, SUMO.

3. Identify the set of concepts required for the domain(s), e.g. building, floor, room. Some possible approaches include top-down, bottom-up or a combination of those (Uschold and Grüninger 1996).

4. Add or modify the concepts from the top-level ontology only if necessary, as this could pose interoperability problems. If a concept is introduced or modified, care should be taken to preserve the interoperability. So, for example, a new concept may be a synonym or a sub-concept of an existing concept. In such cases, the appropriate links between concepts should be included as well.

For Steps 3 and 4, the objective is to identify as many relevant concepts as useful and interoperable in other similar domains. Some guiding questions may include the following:

- Does this concept apply generally in other similar domains?
- Is this concept necessary in conducting reasoning in the domain(s)?

In pervasive computing, this is the ontology that defines the key concepts about any building, including concepts such as floor, wing, level and room and relationships between such as floor is-part-of building or building is-a fixed structure.

3.2 Application Ontology

An application ontology specifies the application-specific vocabulary, which is more specialised than both the top-level and middle ontologies and often cannot be reused (Guarino 1998). At this layer of PECO, the aim is to construct a base ontology by carefully analysing the middle ontology and an authoritative document source, such as a building plan, which would provide a foundation for populating application-specific concepts and relationships. It is important that this process can be automated or semi-automated, to ensures a more complete and systematic approach to model and reason about the highly dynamic pervasive computing environment. One way to do this is to exploit existing document sources.

Since an application ontology is application-dependent, the potential to reuse such an ontology is low. While it is important to account for interoperability and use a standard representation format for the top-level and middle ontologies, it may be acceptable to
model this layer of PECO with a less conventional representation language that is more suitable for modelling and reasoning in that domain and application.

Note that while the ontology created at this application level is not reusable across different contexts, the tools that create it may be. In our case, the ontology that we create captures relevant relationships about one specific building. However, the process and code used to create it from the building maps could be used for any other building for which there is a similar representation in Scalable Vector Graphics (SVG) format. Similarly, a program could be created to operate in conjunction with the building model representation used in the various Computer-Aided Design (CAD) packages that architects use.

3.3 Accretion Ontology

The bottom layer of the PECO model is an accretion ontology; this is key to our personal ontological representational approach. The way PECO accretes evidence from a range of sources to populate an ontology and generate adaptive information depending on the user and context is accretion and resolution, described in Section 2.4. In short, it accretes evidence and resolves a value that is both time and context relevant in real time. We call a function that resolves a value with the accreted evidence a resolver. This approach forms the bottom layer of the model and serves as a bedrock for reasoning about personalised and contextualised information in a pervasive computing environment.

In PECO the accretion step happens when the propositions found from various sources are populated into the base ontology. As each proposition is extracted, it is simply stored as evidence with a timestamp and a source in the relevant concept without further interpretation. The proposition may, for example, be from a sensor detecting a mobile device (e.g. Bob’s mobile phone is in Room 123), a staff directory (e.g. Bob has an email address bob@example.com) or an assertion by a user (e.g. Bob is in Room 123).

3.4 Reasoning with PECO: Ontology- and Evidence-based Context Reasoner

Ontology- and Evidence-based Context Reasoner, or ONCOR (see Figure 3.2) is a reasoning mechanism especially designed for interpreting PECO. It performs the resolution process of the A/R approach using ontological reasoning: when a value is required for a concept in a proposition (e.g. where is Bob?), the stored evidence is examined and
resolved to a value. This process can be personalised depending on the context of the program or person requiring the value. This is done by running a customised resolver function. The following two examples illustrate when using different resolvers may help personalised reasoning.

- For someone who usually carries a mobile phone, evidence for the mobile phone’s location is valuable for determining that person’s location; but for a person who often forgets to carry their mobile phone, the evidence would not be as reliable, and the resolver function may account for that fact. This means that in one case, we can assert that the person is at the place where their phone was detected; in the other case, we cannot.

- Taking another example, Bob may only want to release his location information to the general public at a relatively coarse grain-size, say, at the level of the building. So, he may also choose to exclude evidence generated by his mobile phone. This can be done by using a resolver function which is blind to such evidence when releasing Bob’s location to the general public.

ONCOR essentially consists of an array of resolver functions to be used adaptively to resolve personalised and contextualised information. This value resolution process provides a flexible and pragmatic approach to reasoning with light-weight ontologies of places, devices and sensors in a context-aware system. In the following subsections, we describe representative resolvers that can be used to provide adaptive pervasive computing information.

### 3.4.1 Location Resolvers

Since the key to the reasoning is in the evidence sources and their interpretation by reasoners. We now consider the approaches that might be used to reason about an
ontological aspect in pervasive computing, a person’s location.

We describe six approaches to reasoning about location based on diverse sensor evidence. The first three are representative, non-ontological resolvers commonly used in other location systems, and the next three are ontological resolvers. For each, we first explain the design motivation (for the Point and Time Decay algorithms, we present the motivation to include them), then provide an illustrative example referring to Table 3.1 followed by some discussion of that approach, its expected power and limitations.

**Point.** Motivated by the common case that 
*fresher* location data is more reliable, this approach selects the most recent piece of evidence generated by any sensor (High-tower and Borriello 2001). So, from the list of sensor evidence in Table 3.1, this algorithm would select the value of Level 1 West, as this is the most recent evidence. It is the least computationally expensive algorithm for resolving a location value with sensor data. Its simplicity has attracted many systems using single grain-size sensors, such as commercial infrared badge location systems, as reviewed by High-tower and Borriello (2001). The downside of this approach is that the latest piece of evidence might not be the most desirable one.

**Time Decay.** To accommodate inaccurate sensor readings, fuzziness is often desired; this algorithm is based on the exponential decay theory to weigh each evidence source in relation to their *freshness*. Multiple evidence sources for each unique location value are aggregated in favour of the most weighted evidence source (i.e. 

<table>
<thead>
<tr>
<th>Detection Time</th>
<th>Location</th>
<th>Sensor Type</th>
<th>Detected Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 minute ago</td>
<td>Level 1 West</td>
<td>Bluetooth sensor</td>
<td>Bob’s Mobile Phone</td>
</tr>
<tr>
<td>1 minute ago</td>
<td>Desk 116-01</td>
<td>Login sensor</td>
<td>PC-01 at Room 116</td>
</tr>
<tr>
<td>2 minutes ago</td>
<td>Desk 116-01</td>
<td>Login sensor</td>
<td>PC-01 at Room 116</td>
</tr>
<tr>
<td>3 minutes ago</td>
<td>Desk 3W32</td>
<td>System Activity sensor</td>
<td>Bob’s Office PC</td>
</tr>
<tr>
<td>3 minutes ago</td>
<td>Level 3 West</td>
<td>Bluetooth sensor</td>
<td>Bob’s Laptop PC</td>
</tr>
<tr>
<td>3 minutes ago</td>
<td>Level 1 Middle</td>
<td>Bluetooth sensor</td>
<td>Bob’s Mobile Phone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Resolved Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>Level 1 West</td>
</tr>
<tr>
<td>Time Decay</td>
<td>Level 1 West or Desk 116-01</td>
</tr>
<tr>
<td>Bias</td>
<td>Desk 3W32</td>
</tr>
<tr>
<td>Closest Common Subsumer</td>
<td>The Building</td>
</tr>
<tr>
<td>Granularity Harmoniser</td>
<td>(conflict is detected, but remains unresolved)</td>
</tr>
<tr>
<td>Democratic</td>
<td>Level 1 West and Desk 116-01</td>
</tr>
</tbody>
</table>
the latest evidence source) for each location value as well as the locations with more repetitions. From Table 3.1, this algorithm would probably select the value of either Level 1 West or Desk 116-01, depending on how the repetition of appearance weighs proportional to the time weighting. Equation 3.1 calculates the weighting of each given evidence source (i.e. \( w_i \)) based on their timestamp using an exponential decay function\(^3\), where the adjusted decay constant (\( \lambda \)) is set to be 0.01. Equation 3.2 discounts the adjusted weights of evidence sources of other location values from the maximum weight for the current location value, where \( n \) is the total number of evidence sources, \( m \) is the number of evidence sources of the location value whose weight is being calculated:

\[
w_i = N(t_i) = e^{-\lambda t_i} \tag{3.1}
\]

\[
TimeDecay(w_{1..m..n}) = \max(w_{1..m}) - \frac{\sum_{i=m+1}^{n} w_i}{n} . \tag{3.2}
\]

In Table 3.1 each evidence source reports a different location, so \( m \) would be 1 for each source. If, for example, Level 1 West is reported in another evidence source 3 minutes ago, the more recent one would be used, but this piece of appearance would make \( m \) be 2 and increase the total weighting of the other evidence source reporting the same location. This algorithm would be unfavourable when newer evidence is not more accurate and when there is false positive evidence.

**Bias.** Many location systems would have ad-hoc algorithms specially accounted for their own sensor infrastructures. Motivated by the fact office users tend to spend more time at their own offices/desks (González and Mark 2004), this algorithm selects the user’s personal work space if it is found in the most recent set of evidence sources. Otherwise it returns the most recent evidence source. From Table 3.1, this algorithm would select the value of Desk 3W32, as that is Bob’s work space. Developing an ad-hoc algorithm may be a time-consuming exercise. It may require the algorithm developer to collect a reasonable amount of evidence and to spot system patterns in order to design a suitable algorithm, and this often requires tuning. In many cases, this type of algorithm is not generalisable in another similar domain.

Closest Common Subsumer (CCS). This approach is motivated by the goal of determining the finest grained location that is consistent with a set of evidence. This algorithm uses the location ontology to find the closest common subsuming location: this is similar to the concept of finding the lowest common ancestor in graph theory. From Table 3.1, this algorithm would give the value of the building spatially containing all the location values. This algorithm provides an indication of the level of conflict in data from different sensors; the algorithm can be used to resolve a value for location that is consistent with all available evidence.

Even though this algorithm may provide more accurate location resolution, the resulting precision can be unacceptable. So, for example, when Bob’s presence is inferred by sensors in both Room 100 and Level 3, this algorithm would resolve the location in the building-level granularity; that would be of minimal value for a query that requires a finer-grain location, such as “Where is Bob in the building?”

Granularity Harmoniser (GH). This algorithm is motivated by the common inconsistency of granularity variance seen in multi-sensor networks. So, for example, if Bob’s presence is inferred by sensors in Level 3 and Room 300, GH would return Room 300 as it is contained in Level 3. GH uses a location ontology to determines whether or not a list of locations are all on the direct is-part-of ontological path (e.g. Level 1 West is-part-of Level 1). If they are all on the direct ontological path, it returns the finest-grain location in the list. Otherwise a conflict is reported, and another designated conflict resolver is used to resolve the inconsistency. So, from Table 3.1, this algorithm would not be able to resolve a unique location value. Instead, it would report an inconsistency.

Not only can this algorithm effectively harmonise granularity variance, it also reveals other forms of conflicts in sensor evidence that occur in a multi-sensor environment. Importantly, the simplicity of this algorithm facilitates the critical aspect of scrutability, enabling end applications to provide understandable explanations of this reasoning. Figure 3.3 illustrates how a system can explain to a user how the GH algorithm operates. This algorithm would only be effective if the sensor infrastructure has complete-containment relations (i.e. no overlapping locations) between the locations modelled.

Democratic. The assumption of this algorithm is that accurate data tends to be more common than noisy data. This algorithm groups the location values according to the set of subsuming locations at the same level of granularity (e.g. Level 1, Level2, Level 3). The groups that contain less evidence, or the minority groups,
are discarded. So, from Table 3.1, this algorithm would go through an iterative process of filtering out minority groups by firstly filtering out Desk 3W32 and Level 3 West, as they are both spatially subsumed by Level 3, and has fewer location values than that on Level 1. Then it would filter out Level 1 Middle, as it has fewer location values than Level 1 West. At last, it would give the location values of Level 1 West and Desk 116-01 as the possible candidates. The democratic algorithm may eliminate some of the false positive sensor evidence as described in Section 1.2 on page 7. It may fail if false sensor evidence sources overweight the genuine ones. Algorithm 1 is the pseudocode of an algorithm that uses Democratic and GH as filters, with the Point algorithm as the final resolver if a conflict still exists.

The Granularity Harmoniser and Democratic algorithms are filters: they do not guarantee complete conflict resolution, but try to filter out certain conflicts. For example, GH and Democratic algorithms can be used to filter out conflicting evidence before passing the unresolved evidence to the Point algorithm (see Algorithm 1). Each algorithm may perform well, in terms of both accuracy and precision, in different contexts. When precision is within a tolerable location granularity, CCS may be a good candidate for accuracy. When the deployed sensors include overlapping locations in granularity difference, GH can be a good filter in the process of interpreting the evidence. The Democratic algorithm may help filter out false positive sensor evidence.

The ontological algorithms are alternatives to location resolution, as well as a more systematic and context-independent approach than an ad-hoc algorithm, such as the Bias algorithm. As shown in Table 3.1, each resolver may yield different values and one may perform better than the other in certain context. One step further would be
CHAPTER 3. THE PERSONAF CONCEPTUAL FRAMEWORK

Algorithm 1: An algorithm that adopts both the Democratic and the Granularity Harmoniser approaches

Input : evidenceList, each evidence consists of a timestamp and a location
Output: A location, from an evidence source in evidenceList
/* Non-finest grain-sizes could be a wing, a level, or a building */
foreach grainSize in non-finest grain-sizes in a spatial ontology from coarse-to-fine order do
    groups = group evidenceList by grainSize to a mapping like {'level 1': [e1, e2 ..], 'level 2': [e_n, e_{n+1} ..], .. 'level k': [e_m, e_{m+1} ..]}
    maxSize = 0
    foreach group, evidenceSources in groups do
        maxSize = max(|unique locations in evidenceSources|, maxSize)
    endforeach
    foreach group, evidenceSources in groups do
        if |unique locations in evidenceSources| < maxSize then
            remove group from groups; /* remove the minority */
    endforeach
    evidenceList = evidence sources for each group in groups
    /* filter out granularity-variance conflicts */
    resolvedLocation = granularityHarmoniser(evidenceList)
    if resolvedLocation ≠ null then
        return resolvedLocation
    else
        return point(evidenceList)
    endif
endfor


3.4.2 Other Resolvers

A number of other types of resolvers are also used for delivering personalised information. These include resolvers for inferring social connections (e.g. friends, colleagues), for finding places the user has been to, and for obtaining less dynamic demographic information, such as name, gender, and office location of the user. There is a set of generic resolvers from the PersonisAM framework (Assad et al. 2007), such as MostRecentEvidenceValue
and \textsc{RecentUniqueValues} resolvers, allowing more sophisticated resolvers to be built rapidly and flexibly. The generic resolvers that were used as part of the ONCOR resolvers are listed in Appendix B on page 136. Here we describe two representative resolvers to demonstrate how a resolver function may be used with an ontology to interpret accreted evidence of propositions in PECO.

**Social Connection Resolver.** This resolver function is to determine social connections of a user, as well as the reliability for each relationship. To illustrate the point, we use arbitrary weighting for each evidence source and assume three relationships a user may have with other users: friends, family members, and colleagues. If we are only interested in discrete social relationship (e.g. A is either a friend or not a friend of B), the \textsc{RecentUniqueValues} resolver, which finds unique evidence values, would suffice. To obtain fuzzy confidence weighting, which is more realistic and often more desirable, the evidence source would also need to be examined for weights to be calculated. For example, one colleague-relationship is obtained by mining internal email lists of a company; another colleague-relationship is input manually by the Human Resource team. The latter would be intuitively deemed more reliable.

**Been-to Places Resolver.** This resolver finds all places that a user has been to within some period of time at a certain location granularity. So, for example, it may list the places that Bob has been to in the last month as Level 1 East, Level 1 West, and Level 3 West, or at a coarser granularity: Level 1 and Level 3. To do this, we first need to collect the list of location evidence of Bob in the last month. This is done using the generic resolver that fetches the list of evidence for a particular relation (e.g. Bob’s location) given a length of time. Then the locations are normalised to the desirable granularity using a location ontology, such as the SIT ontology to be described in Section 4.2 on page 51. This resolver is used to determine the user’s knowledge about the SIT building when deciding whether to deliver a personalised place label, such as Bob’s office.

Now consider applying the resolvers above in the example introduced in Figure 1.1 on page 4, which reappear as Figure 3.4. It is then possible to reason about the following:

- David is in Room 300, inferred from evidence of his activity sensor using a location resolver.
- Room 300 is David’s office, inferred from evidence mined from the staff directory using the \textsc{MostRecentEvidenceValue} resolver.
• Bob is a colleague of David, inferred using the Social Connection resolver.

• Bob has been to Room 300 in the last month, inferred using the Been-to Places resolver.

• David is a male, inferred from evidence mined from the staff directory using the MostRecentEvidenceValue resolver.

These in turn allow the system to infer personalised location information, such as David is in his office presented in Figure 3.4. Moreover, system scrutability is supported by this reasoning process; understandable explanation of this reasoning process can be generated.

3.5 Summary

We have presented a conceptual framework called PERSONAF that encompasses a personal ontology (PECO) and a reasoning component (ONCOR). PECO provides great flexibility in terms of reasoning and self-evolving, both essential properties for a semantic knowledge base in pervasive computing, a dynamic environment, overloaded with information.

PECO is based on a three-layer model:

• A middle ontology layer for computational efficiency and reusability. It normally reuses a top-level ontology, such as OpenCyc, in order to interoperate with a broader range of domains.

• An application ontology layer for describing application-specific vocabulary. A well-defined application ontology may form a base ontology for semi-automated ontology population, which, in turn, facilitates a self-evolving knowledge base. The trade-off is its relatively low interoperability and reusability.
• **An accretion ontology layer to provide personalised and contextualised reasoning.** It is built based on the accretion and resolution approach, which allows the ontology to be populated with a range of heterogeneous sources. This novel approach forms the bottom layer the of the PECO model.

Following the A/R approach, we introduced ONCOR that consists of a number of resolver functions, or simply *resolvers*, to resolve a range of personalised and contextualised information. We presented six location resolvers, a focus of this thesis, with three representative, non-ontological algorithms and three ontologically-based algorithms. In addition, we demonstrate the flexibility and power of the A/R approach by describing two of the more sophisticated resolvers that utilise various generic resolvers in PersonMIsAM.

Having introduced this powerful PERSONAF conceptualisation, we present the process of implementing such a framework in an indoor pervasive computing environment in the next chapter.
Chapter 4

Using PERSONAF in Pervasive Computing\textsuperscript{1}

In this chapter, we describe how we implemented the PERSONAF framework by creating each layer of PECO, as well as how ONCOR may reason with PECO about personalised information in pervasive computing.

Figure 4.1 illustrates what a concept in PECO would look like; it shows what propositions of a particular room, Room 125 may have. The underlined text under each concept indicates the evidence sources from which the concept/relation pair was derived. So, for example, the proposition of Room 125 is-a common room is derived from both the technical building data and a building manual. So this applies, generally, to most people. By contrast, the proposition Room 125 is-a social hub is extracted from a postgraduate student handbook and, therefore, only applies to postgraduate students. The proposition Room 125 is-a recharging corner is given by Bob and, thus, it is only valid for Bob himself.

This chapter starts by describing the construction of a middle ontology (i.e. the top layer of PECO) for the inside of a building from a top-level ontology. Then we explain the process we followed in constructing an application ontology (i.e. the second layer of PECO) to serve as a foundation of the bottom layer of PECO: this is an accretion ontology, which is populated with propositions mined from application-specific sources. Finally, we present an evaluation of this ontology based upon a comparison of PECO’s reasoning power with the two other most similar context ontologies available: COBRA-ONT and CONON. More extensive evaluation in terms of the power of the ontology for reasoning about conflicting location evidence and for providing personalised information

\textsuperscript{1}Parts of this chapter have been published in the conference proceedings of IUI 2007 and AH 2008 in a shorter form.
4.1 MIDDLE BUILDING ONTOLOGY

As described in Section 2.1 on page 14, there have been several spatial ontologies built for pervasive computing, including, for example: the space ontology in SOUPA (Chen et al. 2004e), which models geo-spatial coordinates for various types of geographical regions, as well as spatial subsumption relations between them; the location ontology in CONON (Wang et al. 2004b), which has a two-level location ontology with the root concept being Location and two sub-concepts, OutdoorSpace and IndoorSpace; and the spatial ontology in UbisWorld (Heckmann 2006), which is a blend of location and activity ontologies describing purposes for a location.

None of these, however, provides a suitable upper/middle ontology to extend upon for reasoning about the inside of a building. The differences in application focus and scope often result in an over-generalised location ontology for our system. For example, CONON’s two-level location ontology is a general upper ontology for a wider region, whereas we need a general ontology for indoor space. Thus, we built a Middle Building Ontology (MIBO), as the top layer of the PECO (see Figure 3.1 on page 35), for modelling locations in a building and for reasoning about location information in a pervasive computing environment.

MIBO captures a general structure of indoor space and it is important to account
for reusability and interoperability. For these reasons, we express MIBO with the Web Ontology Language, or OWL, and it is built using a widely available ontology editor, Protégé\(^2\). Like many other manually constructed ontologies, MIBO has undergone an iterative process similar to the knowledge engineering methodology described in (Noy and McGuinness 2000): from choosing competency questions to narrow down the scope, to reusing existing ontologies and defining concept hierarchies and relationships. Competency questions are those that the goal ontology should be able to answer. Below, we discuss how we use some of the competency questions as part of an evaluation of MIBO (Section 4.4). In the rest of this subsection, we describe design decisions made in the choice of ontology representation and the existing ontology to use as a foundation for MIBO.

### 4.1.1 Ontology Representation

Two critical properties an ontology should possess are reusability and interoperability: using an interoperable language is essential to facilitate and encourage both qualities. OWL is compatible with some other popular ontology languages like SHOE (Heflin and Hendler 2000) and DAML+OIL (Horrocks 2002). The fact that OWL is a World Wide Web Consortium (W3C) standard earned it popularity in the ontology community. Like many other meta languages, it is not very intuitive and is hard to use (Pulido et al. 2006).

But OWL editors, such as Protégé, SWOOP\(^3\), and SemanticWorks\(^4\), help overcome these shortcomings. In addition, its several logic relations support automatic inferences (mainly for OWL-Lite and OWL-DL; OWL-Full may be undecidable for some queries).

We also considered the Simple Knowledge Organisation System (SKOS) extension, a developing Semantic Web standard in W3C. It provides a simple yet relatively expressive set of relationships to describe concepts, especially for thesauri and dictionaries. Its use has been demonstrated in a number of domains (Isaac et al. 2007; Kay et al. 2006). However, its lack of reasoning mechanisms and unstable standard status made OWL preferable.

---

\(^2\)http://protege.stanford.edu/, 12 Sept 2006
4.2. APPLICATION ONTOLOGY

4.1.2 Reusing Existing Ontologies

Most of the concepts, as well as their hierarchical relations in MIBO are borrowed from OpenCyc (Lenat 1995). Other sources we considered are WordNet (Miller 1995), ConceptNet (Liu and Singh 2004) and ThoughtTreasure (Mueller 1998). They are all relatively large databases that contain various concepts in English; WordNet has more than 200,000 word-sense pairs, OpenCyc and ConceptNet both have around 300,000 concepts, and ThoughtTreasure about 27,000 concepts.

ConceptNet and ThoughtTreasure structured their ontologies for natural language processing in common sense reasoning, and neither of them provides their ontologies in OWL or any other more general ontology language: a disadvantage for interoperability. WordNet is a semantic lexicon, mainly designed for lexical analysis rather than spatial relations.

OpenCyc, which is a subset of Cyc, is a more general database that aims to describe real life knowledge. As a result, it has a comprehensive coverage of spatial entities and hierarchical relations about space and location that best suit our needs in defining an ontology for buildings, where the goal is to support location modelling and applications which make use of this. Other projects have also adopted OpenCyc as a foundation for building a spatial ontology including, for example, SOUPA (Chen et al. 2004e) and the system by de Freitas Bulcão Neto and da Graça Campos Pimentel (2005). The top rectangle in Figure 4.2 shows the MIBO. So, for example, this has the proposition of Building is-a FixedStructure. Appendix A.1 on page 128 lists the full MIBO represented in OWL.

4.2 Application Ontology

An application-specific ontology for an indoor space needs to be manually or semi-automatically built according to the layout of each modelled space. This forms the second layer (i.e. application ontology) of the PECO model. Ideally this ontology for the School of Information Technologies building on the University of Sydney campus would be automatically constructed from building plans, such as that shown in Figure 4.3(a), but there was no straightforward means to convert the proprietary CAD file format into a desirable format. Hence a hand-crafted, to-scale version was built (see Figure 4.3(b)).

---

5 We used the OWL version of OpenCyc v0.7.8b http://www.cyc.com/2004/06/04/cyc, made available on the OpenCyc website.
7 MIBO is available in OWL online at: http://www.it.usyd.edu.au/~niu/ontologies/2008/06/mibo.
An XML-based form of SVG was used to represent the SIT maps. SVG is an industry standard, W3C specification describing text and graphics in XML. This format was chosen for its flexibility, being a lossless image format, simple editing as it is in plaintext XML syntax, and interoperability as it is a W3C standard.

With the SVG maps, we were able to semi-automatically build a spatial ontology, which extends the MIBO, for the SIT building, which we call SIT ontology. Figure 4.2 shows an example portion of entities in the SIT ontology that extends the MIBO, at the top of the figure. Each unlabelled arrow represents an is-a relationship. There is no semantic difference between solid and dotted lines; dotted ones are only for visual clarity. So, for example, Level 3 (to the left below MIBO in Figure 4.2) both is-part-of SIT Building and is-a LevelInABuilding. Appendix A.2 on page 132 shows the SIT ontology represented in OWL.

First, we note that the ontology has two parts: a handcrafted, general ontology for buildings (Figure 4.2 top) and a generated part that captures the structure of a particular building (Figure 4.2 bottom). A key property of MIBO is the careful separation of the costly handcrafted but reusable parts. For the reusable parts, we make use of automated analysis of information about buildings to generate the remainder of the ontology for a new building and to make it easy to handle the addition of new concepts and instances, such as adding a new sensor. It is designed to provide a flexible and pragmatic approach

---

8 The application ontology is available in OWL online at: http://www.it.usyd.edu.au/~niu/ontologies/2006/09/soit-building.
4.3. ACCRETION ONTOLOGY

The SIT ontology is further populated with propositions mined from multiple sources, such as a building manual and sensor data. Each source constitutes different degrees of

to building a light-weight ontology by allowing concept and relation population from various document sources on an application ontology.

“Every [knowledge] representation is a trade-off between expressiveness...and efficiency...” (Lenat 1998). While the Web Ontology Language OWL is a popular format for representing ontologies, it is, after all, not designed for modelling entities in pervasive computing environments. Moreover, the limited reusability and interoperability of an application ontology, as discussed in Section 3.2 on page 37, makes OWL less attractive than a format that is more suitable for reasoning about that particular application. We adopt the accretion-resolution approach and PersonisAM (Assad et al. 2007), a suitable framework for this approach, to model the SIT ontology.

4.3 Accretion Ontology

The SIT ontology is further populated with propositions mined from multiple sources, such as a building manual and sensor data. Each source constitutes different degrees of
reliability and relevance in different contexts. A proposition can contain an arbitrary number of attributes, which typically include a source, where the proposition is mined, and a timestamp. Other possible attributes may be a value representing reliability and an expiration date. So, for example, a proposition of *Room 123 is a social hub* may be from the 2008 postgraduate student handbook (i.e. the source) and is mined in 1 January, 2008 (i.e. the timestamp). The mined propositions can be roughly divided into five classes: stereotypes, facts, observations, given information, and inferences. In the following, we describe what implications each class of proposition has in terms of reasoning, the sources that are and can be used for each class, our approach to each source in populating the ontology, and how the mined propositions affect personalised reasoning. Those are then followed by a discussion on the expected reliability of each class of proposition and how the propositions contribute to useful inferences in a context-aware system.

### 4.3.1 Stereotypes

This type of proposition helps model users with stereotypes, similar to those described in (Rich 1979). However, the reasoning mechanism is quite different in that we allow for the accretion of a range of evidence for all sources that appear to be relevant to an individual. Stereotypic values are defaults and typically do not apply to all users. Instead, they are expected to be overridden when more reliable information become available, such as an observation or user input (Kay 1994). So, for example, there may be documents written for particular groups of the users, such as a postgraduate student handbook.

One source we use that contributes, in some aspects, to this type of proposition is the technical building data (see Table 4.1). It shows a sample view of extracted information from the technical building data. The first row shows that Room 125 is a common room, and the usual activities conducted there are listed as casual breakout; it is accessible for all staff and should be next to the boardroom. For example, using only evidence extracted from technical building data, Room 125 may be introduced to administrative staff in an orientation tour, but not Room 203, as it is only relevant to academic staff. Therefore, in the context of personalisation, the proposition of *Room 203 is a Pervasive Computing Lab* is a stereotypic proposition that is relevant only to academic staff. In our case, the building is new and we were able to make use of the set of as-built documents, written by the builders, consultants and architects, handed to the client on building completion. One down side is the lack of format of the information, hence less machine readable. Even though some information still needs
4.3. ACCRETION ONTOLOGY

Table 4.1: A sample view of extracted technical building data

<table>
<thead>
<tr>
<th>Rm No.</th>
<th>Room Name</th>
<th>Principal Activities</th>
<th>Primary Users</th>
<th>Essential Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>common room</td>
<td>casual breakout / eating</td>
<td>all staff</td>
<td>boardroom</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and drinking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>IT boardroom</td>
<td>meetings / presentations</td>
<td>all staff</td>
<td>directly adjoining</td>
</tr>
<tr>
<td></td>
<td>kitchenette</td>
<td></td>
<td></td>
<td>boardroom and common</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>room</td>
</tr>
<tr>
<td>131</td>
<td>programming staff</td>
<td>1 x office</td>
<td>programming staff</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>pervasive computing lab</td>
<td>experimental space</td>
<td>academic staff</td>
<td></td>
</tr>
</tbody>
</table>

human intervention before actually populating the propositions, this class of documents is still of great potential for integration of pervasive computing facilities.

Another source we use is internal email aliases. This type of information can give evidence about the social connection of two people. The assumption is that people who are on the same group email alias list know each other. We make use of the mail aliases, such as those used to announce weekly seminars of research groups. This is useful, though not always correct; for example, a new student may well come to know the academic staff more quickly than the other way around. Nonetheless, this provides an easy way to gather evidence about workplace social network.

Other sources that could be mined for evidence of stereotypes from a variety of sources. We now briefly describe these:

Postgraduate/undergraduate Student Handbooks

The content of a student handbook can vary between institutions, but there is often valuable information that can be used for context reasoning about the students. For example, the Postgraduate Survival Manual of the University of Sydney lists facilities (e.g. printers, computers), services (e.g. counselling service, learning centres), financial assistance, and other information relevant to postgraduate students. The mined content should only be assumed to be relevant to the respective students, since they are the intended audience for such documents.
Department/faculty Information about Students or Staff

This type of information can often be found as both hard copy (e.g. handouts during orientation) and soft copy, such as on the departmental/faculty websites. Similar to student handbooks, it typically describes various policies, services, facilities, and contact information that a student would need to know about, but it targets smaller groups of students from the respective department/faculty.

University Calendars and Timetables

Many universities have a calendar listing schedules of each lecture theatre and meeting room and events held. Some departments and faculties would provide a timetable of courses, including the time, the place, the lecturer, and even a summary of a particular lecture. Some of the information may be targeted to certain groups of people. For example, information about a lecture may only be relevant to the students enrolled in that particular course. This may be combined with the user’s personal timetable to enable context-aware information delivery.

4.3.2 Facts

Facts are the type of propositions that are generally true for all users under most situations. One example is the spatial containment relationship between a room and a floor in a building. The sources are normally authoritative and are rarely updated. The ones we used include the building plans and technical building data, to be described next. Another source that can be mined is a building manual, which describes various aspects about the building. So, for example, it contains information about who and where to go for an after-hours access card, and available facilities (e.g. microwaves, refrigerators) at particular places. This would, however, require more sophisticated natural language processing than simple parsing of formatted languages.

Building Plans

Building maps (Figure 4.3(b)), which are created from the building plans, are used for getting spatial relations about the building. Figure 4.3(a) shows the building plans of a floor in the SIT building. We use computational geometry to extract relations such as adjacent-to, nearby, and has-area. We can establish informal definitions of these concepts and these indicate how a map might be used to extract such relationships in a building. Adjacent-to is defined as when two locations are physically connected to each other (see Figure 4.4), by having touching walls (Figure 4.4(a)), touching corners (Figure 4.4(b)),
4.3. ACCRETION ONTOLOGY

Figure 4.4: Possible situations of adjacent locations

(a) Touching walls

(b) Touching corners

(c) A wall touching a corner

There are valuable applications for such spatial relations. For example, they can be used to model the detecting range of a sensor in terms of the locations within some set distance: if a Bluetooth sensor is installed in Room 125, with the nearby spatial relation, it is possible to infer the rooms that a detected device is probably at. A very different example is to infer likely social connections between people who have offices close to each other. For example, if Alice has a desk next to Bob’s, they are likely to know each other.

One potential problem is that spatial distance on the map may not correspond to actual walking distance. As illustrated in Figure 4.3(b) on page 53, although Room 126 is physically adjacent to both Room 131 and Room 125, the walking distance between Room 126 and Room 131 (i.e. the white line connecting the black dot and the black triangle) is much longer than that between Room 126 and Room 125 (i.e. the white line connecting the black dot and the black square).
Technical Building Data

Again, referring to Table 4.1 on page 55, possible uses for the extracted information include: obtaining a more human-understandable name for a room (e.g. *common room* instead of Room 125); better location estimation and way-finding by knowing nearby locations; the “Primary Users” information may help systems deliver personalised information to different groups of users. The weakness in this source follows because it was designed to be read by construction personnel and some names/values are underspecified: e.g. *programming staff* as the “Room Name” for Room 131.

4.3.3 Observations

This type of proposition is contextual information *observed* by machines or devices, which are typically called sensors. Sensors can be pragmatically divided into two classes: physical and virtual (Indulska et al. 2003). Physical sensors are used to detect contextual changes in the physical world, such as physical movements, sound, and light. Virtual sensors, on the other hand, are software programs used to extract contextual information from the virtual space, such as mouse and keyboard activities, network connection, and user login/logout. There are six types of sensors deployed in the SIT building. The first three types have been described in detail in (Carmichael, Kay, and Kummerfeld 2005).

**Bluetooth Sensors.** For sensing in the range of roughly 10 metres, they consist of Bluetooth dongles that periodically scan for surrounding Bluetooth-enabled devices and report a list of detected devices. They occasionally fail to pick up some signals in range and may suffer from overlapping detection. People need to register their Bluetooth-enabled devices if they want these to contribute to their user models.

**Activity Sensors.** For sensing at the desk/room grain-size, they are programs users can choose to install on their machines, so that the sensor can report their activity/inactivity based on keyboard and mouse movements and (de)activation of screensavers. One assumption is that each user would not share their account with another user, which would fail when they do share their own account with a close friend or a family member.

**Login Sensors.** For sensing at the desk grain-size, they are programs that monitor users logging in and out of machines in computer laboratories. Since the machines in the laboratories are shared by many people, users normally take extra care in
logging off when they are physically away, hence reliable readings of users’ presence in the laboratories.

**Infrared Sensor.** This sensor is triggered by an object or a person moving in its light-of-sight, which is set up to be about one meter just inside of a door. This would enable anonymous detection of people moving in and out of an area through the door. Its weakness is the inability to determine multiple people passing by in a group.

**Reed Sensor.** The reed sensor is used to determine if a door has been opened or closed. When deployed with an infrared sensor, it is possible to determine whether a person is entering or leaving an area through a door by examining the order in which the sensors are triggered. If the infrared sensor—installed just inside the door—is triggered before the reed sensor, that typically means a person is leaving that area. Together with another sensor that identifies the users, this is used to determine when a user comes in or leaves an area. The weakness of this type of sensor is that an arbitrary number of people may pass through an opening door and that will only send one signal as the door is closed.

**Nike+iPod Sports Kit.** This kit comes with a sensor/transmitter, designed to be put inside a person’s Nike+ shoe, and a receiver to be connected to that person’s *iPod Nano* device, so that person can receive audio feedback about their movements (e.g. distance travelled, calories buried) when they walk or run (Saponas et al. 2006). Each Nike+iPod transmitter has a unique identification, so it is possible for a Nike+iPod receiver to distinguish a number of the transmitters. This is used as another readily available proximity sensor to the Bluetooth sensor. The Nike+iPod transmitter emits radio frequency signals in the range of 10–20 metres similar to Bluetooth sensors, but the signal is less penetrative (e.g. through walls). This also requires the user to register their own Nike+iPod transmitter. The fact that the Nike+iPod transmitter only emits signals when enough force is applied to it (e.g. by moving or jumping), it may fail to wake up if a person walks too softly. On the other hand, if a person runs pass a Nike+iPod receiver—hence generating enough force for the transmitter to send out signals—it is very probable for the receiver to pick up the signals.

---

4.3.4 Given Information

The sources we used for this type of proposition are staff and student directories, device and sensor information, and direct user input. Staff and student directories contain valuable information. For example, they help provide more human-understandable names for personal offices and workspaces. The roles for staff (e.g. lecturers) and students (e.g. full-time PhD students) can also facilitate personalised information delivery. For example, a reminder for a board meeting may be valuable to academic staff, but would be irrelevant to students. Some of the given information, such as a staff directory, can be processed with a simple parser; others are manually input by users (e.g. updating their current location) or system administrators (e.g. sensor information).

A key aspect of our approach is that the system must deal with inaccurate information. For example, the staff directory details of people’s work roles might be out of date, and people may change office. In such cases, users should be able to scrutinise their own models and correct the information, or they may choose to indicate their current location via a map-based interface when they see an erroneous sensor information (see Figure 4.5).

Device and Sensor Lists

Before inferences of location information about users and devices can be made, information about the sensors and devices, such as their MAC addresses and their locations,
needs to be obtained from users and system administrators, and then populated. This source is reliable, although some information may need updating more often than others. So, for example, the person who carries a mobile phone may change more often than the MAC address of a Bluetooth device.

It is also desirable to quickly add additional devices or sensors into the system in operation (see Figure 4.6). Figure 4.6 shows a Web interface allowing system administrator to easily register a sensor, device, location, and user.

### 4.3.5 Inferences

Some additional reasoning needs to be done to some propositions to infer more useful information. This forms another type of proposition, *inferences*. For example, *observations* use some given information to infer the useful propositions of *PersonA isLocatedAt PlaceX*. Inferences are unreliable in nature; the reliability depends largely on underlying information used in the inference resolution.

From the sensors, we eventually want to have evidence in the form of *PersonA isLocatedAt PlaceX*. Physical and virtual sensors produce such inference through different paths of reasoning. For a physical sensor, it would produce simple observations in the form of *SensorA detects DeviceX*. Then, with the location of the sensor (i.e. *SensorA isLocatedAt PlaceX*) and the carrier of the device (i.e. *PersonA carries DeviceX*) known, it is possible to infer the desired proposition:

![Figure 4.6: The Web interface for registering a new Bluetooth device to the Locator system](image)
CHAPTER 4. USING PERSONAF IN PERSVASIVE COMPUTING

Observations (sensor information):

\[ SensorA \text{ detects } DeviceX \] \hspace{1cm} (4.1)

Given information:

\[ SensorA \text{ isLocatedAt PlaceX} \] \hspace{1cm} (4.2)

\[ PersonA \text{ carries } DeviceX \] \hspace{1cm} (4.3)

Inferences:

\[ SensorA \text{ collocatesWith } DeviceX \text{ (from 4.1)} \] \hspace{1cm} (4.4)

\[ PersonA \text{ collocatesWith } DeviceX \text{ (from 4.3)} \] \hspace{1cm} (4.5)

\[ DeviceX \text{ isLocatedAt PlaceX} \text{ (from 4.2 and 4.4)} \] \hspace{1cm} (4.6)

\[ PersonA \text{ isLocatedAt PlaceX} \text{ (from 4.5 and 4.6)} \] \hspace{1cm} (4.7)

Whereas, a virtual sensor would produce observations of \textit{SensorA detects PersonA} (e.g. from an active keyboard and/or mouse). Knowing the machine where the sensor is installed (i.e. \textit{SensorA isInstalledAt MachineA}) and the location of the machine that runs the sensor (i.e. \textit{MachineA isLocatedAt PlaceX}), we can infer the location of the person:

Observations (sensor information):

\[ SensorA \text{ detects } PersonA \] \hspace{1cm} (4.8)

Given information:

\[ SensorA \text{ isInstalledAt MachineA} \] \hspace{1cm} (4.9)

\[ MachineA \text{ isLocatedAt PlaceX} \] \hspace{1cm} (4.10)

Inferences:

\[ SensorA \text{ collocatesWith MachineA (from 4.9)} \] \hspace{1cm} (4.11)
There is a wide range of indoor location sensors with varying accuracy and precision (Chen and Kotz 2000; Hightower and Borriello 2001). It is feasible to integrate additional sensors into our system, making use of a reasoning path similar to one of the two described above. For example, Pink et al. (2008) incorporated a Nike+iPod sensor to the system.

While the example sensor inferences presented above can make use of some logic, such as first-order logic and description logic, there are some inferences that rigid logic would not satisfy, such as social networks and everyday human interaction. The latter requires an understanding of human consensus in real life, or common sense (Lenat 1998), and therefore would be difficult to, or simply can not, be expressed in a logic language. Thus, in this thesis we resolve inferences by ways of active components; this is, associating triggers or rules with the appropriate components (Assad et al. 2007). So, when a component being observed is updated, i.e. a new observation comes in, an inference function would be triggered to deduct relevant inferences with the new information.

4.3.6 Discussions

Those propositions type are borrowed from the evidence types defined in Personis, as described in Section 2.4 on page 31. It should be noted that one source document may provide different types of propositions. For example, the technical building data is mined for propositions of both stereotypes and facts.

In terms of reliability, we normally consider that given information is the most reliable source, followed by facts. Stereotypes and observations, while less reliable, may still provide valuable evidence for personalised reasoning. So, for example, while it may be perfectly correct to call Room 125 a common room, Bob might prefer to call it a coffee room, and his personalised display should reflect that. On the other hand, inferences normally cover a wider range in reliability, as the accumulated deviation increases with more layers of induction.
4.4 Preliminary Evaluation of PECO

Section 2.1 on page 14 describes six common approaches to evaluate an ontology. Here we evaluate PECO with a scenario-based approach with the focus on the syntactic layer, hierarchy, and the application layer.

4.4.1 Syntactic Validation

Syntactic validation is a basic step of evaluating an ontology. With the help of available systems, we were able to validate the syntax and hierarchical consistency of MIBO and the SIT ontology. In particular, the WonderWeb OWL Ontology Validator\textsuperscript{10} had been used for OWL syntactical validation. Both MIBO and the SIT ontology have been written in OWL-Lite and had been successfully validated. In addition to the WonderWeb validation, the ontologies also passed the Ontology Tests in Protégé. The sanity tests check for hierarchical consistency of the ontologies.

4.4.2 Scenario- and Comparison-based Evaluation

We evaluate PECO by answering competency questions we expect the ontology to answer, as suggested in (Noy and McGuinness 2000). The evaluation is further extended by comparing our answers against those of the two most similar context ontologies: CONON (Gu et al. 2004; Wang et al. 2004b) and COBRA-ONT (Chen et al. 2004a,d,e). The answers for both systems are based on the descriptions of the ontologies provided in (Gu et al. 2004; Wang et al. 2004b; Chen et al. 2004a,d,e).

Where is Graf (an academic), asked by Nadal?

Case 1 Nadal is an undergraduate student and is currently in the SIT building. Graf is currently working at a computer in her office, Room 200.

PECO: Graf is in her office.

COBRA-ONT: Graf is in Room 200.

CONON: Graf is in Room 200.

PECO gives a more natural description for Graf’s location: her office. This is based on several evidence sources collected about Nadal (see Section 3.4 on page 38), such as Nadal has been to Graf’s office before, so he knows where that is. In addition, PECO also allows Nadal to scrutinise the reasoning process, i.e. how and why does the

\textsuperscript{10}http://www.mygrid.org.uk/OWL/Validator, 11 Dec 2007
system deduce that Graf is in her office. PECO, however, does not model the quality of context information, as CONON does. So, CONON is potentially able to provide the quality of the information in terms of accuracy, resolution, certainty and freshness. For example, the quality information for the context given above could be that, the location information is given two minutes ago in terms of coordinates with a resolution of 10 metres and an accuracy of 80%.

**Case 2** Nadal is Graf’s son. *He does not know where the Smith Lecture Theatre is, but he knows the Common Room. Graf is giving a speech in the Smith Lecture Theatre, Room 123.*

**PECO:** Graf is in Smith Lecture Theatre (Room 123), which is part of Level 1 East and is nearby Common Room.

**COBRA-ONT:** Graf is giving a speech in Room 123, which is part of Level 1 East.

**CONON:** Graf is giving a speech in Room 123, which is part of Level 1 East.

In addition to spatial subsumption relation, PECO also models other spatial relations between locations, such as adjacent-to and nearby, and these may help form natural direction-telling instructions. It does not yet support activity modelling though.

**Case 3** Nadal is Graf’s son. *Graf’s presence is detected by both sensors in Room 123 and Level 3.*

**PECO:** Graf is in Smith Lecture Theatre. [Depending on which location resolver (see Section 3.4.1 on page 39) is used, the resulting location may differ.]

**COBRA-ONT:** Graf’s location is in Room 123. [Depending largely on the assumption-weighing heuristics, the context broker presents the more reliable assumption about where the user is (Chen et al. 2004d; Chen 2004).]

**CONON:** Graf is in Room 123.

COBRA-ONT and CONON decide the contextual information by using backward-chaining and forward-chaining rules in first-order logic. PECO uses ontological reasoning to resolve conflicting contextual information. PECO also models multiple contextual propositions (e.g. *Room 123 is-a Smith Lecture Theatre* and *Room 123 is-a seminar room*), and inferred that the proposition *Room 123 is-a Smith Lecture Theatre* is relevant for Nadal, the general public.
4.4.3 Discussion

From the expected responses from systems adopting respective ontologies, PECO not only adapts the building ontology for the reasoning, but accounts for each individual’s personal ontology (i.e. what information is relevant to this user), as well as the evidence the user already knows about it (e.g. this user has been in that place). That enables us to analyse the following:

- **What information is relevant to the user at a particular context?** The system should be able to present only the information relevant to a user; this may depend on the user’s access permission level, surrounding context (e.g. what is close to her), and the user’s awareness (e.g. do not present known information).

- **Report why the system believes this piece of information is relevant to me?** Allow the users to scrutinise the reasoning process.

- **Who has queried me (i.e. my location and information about me in general) and what responses were given?** This is important for two reasons: to verify that the system gives reasonable responses; and to provide user control over their own privacy and user model.

- **What does this piece of information presented to me mean?** If the system presents a piece of unexpected information to a user, the user should be able to find out what it means. Taking Case 2 on the previous page, if the system fails to determine that Nadal does not know where Smith Lecture Theatre is, it should be able to present its reasoning in terms of relevant propositions from the ontologies: Room 123 *is-a* Smith Lecture Theatre; Room 123 is *part-of* Level 1 East.

4.5 Summary

In this chapter, we have explained how we have implemented an operational prototype for PECO, which is designed to provide important aspects of reasoning with ontologies in the pervasive computing environment: conflict resolution for sensor fusion, personalised information delivery, and knowledge scrutability. There are three layers of construction for PECO. First of all, a MIBO was built as a middle ontology for the indoor space of a building. Then an application ontology, SIT ontology was built as a base location ontology by extending MIBO. The ontology is built semi-automatically by carefully analysing the SIT maps represented in SVG format. Finally, we described how we constructed an accretion ontology by populating the application ontology with propositions mined from
a range of sources. The propositions which can be one of the five classes: stereotypes, facts, observations, given information, and inferences.

The middle ontology is generally application-independent and should be designed with the concern of interoperability and reusability. At the same time, this hand-crafted middle-level ontology is somewhat specialised in that it includes the concepts and relations that the designed product will be sufficient for the long-term applications for which the ontology will provide a foundation. So, we would expect that the middle-level ontology will support portability to new pervasive computing contexts, such as other buildings, enabling the applications created for one building to readily be ported to work in relation to another building. However, if there are new applications, which need to model additional concepts and relations, this level may need to be revised for them to operate.

The bottom two layers are domain- and application-specific and typically cannot be reused. However, part of the engineering process may well be interoperable so that it is straightforward to rebuild this level of the ontology for a new building. Being able to automate the knowledge engineering process is essential to conduct reasoning in a pervasive computing environment, as it often requires a large amount of domain knowledge to provide effective reasoning.

If, for example, our School were to move to a new building, we should be able to directly reuse the MIBO. In addition, if we had access to similar SVG maps, we should be easily able to automatically create the application ontology for that new building. Some of the documents and resources we use to populate the accretion layer would be directly applicable in the new building. In addition, once the other documents were updated to provide their information for the new building, as would need to be done, we could rebuild this layer from those documents.

On the other hand, one critical aspect that the bottom two layers facilitates is scrutability, the ability to see the reasoning process of the system. The example presented in Section 1.2 on page 11 about system explanation illustrates this potential.

A preliminary evaluation of PECO has been presented, through a validation at the syntactic and hierarchy layers as well as a scenario- and comparison-based evaluation against two other most similar context ontologies. Our work is distinguished by our particular concern for scrutability and support for user control, a particularly important issue for pervasive computing.
Chapter 5

Location Conflict Resolution

In this Chapter we explore the power of the top two layers of PECO—the middle ontology, MIBO and the application ontology, SIT ontology—to support one of the key challenges of modelling locations: the managing of multiple evidence sources, with varying levels of reliability, range, and characteristics that may potentially lead to conflicting information. To achieve this, we needed to obtain a gold standard of movements of people inside a building, so that this could serve as a control to evaluate ontological algorithms for location conflict resolution.

Different approaches to location resolution with diverse evidence sources from sensors are tested in a field study with eight users over a median of 11 days. Our hypothesis is that not only can an ontology facilitate location reasoning at different levels of location granularity, but it can also provide a generic and semi-automatic process for location resolution and noise reduction.

5.1 Experimental Procedure

The study took place in the School of Information Technologies building on the University of Sydney campus. It ran for 10–13 days with a mean of 11 days for each participant. There was an exception for the author participant who had 34 days of logged data. Before the experiment, each participant was given a tutorial on how to use a Tablet PC, the Location Log (LocLog) system and the LocLog Feedback system. The first week was lab trial, for the participants to familiarise themselves with the systems and the logging practice. The data collected during this week was excluded from the analysis.

At the end of each day, the participants were reminded by emails to verify their

\footnote{A shorter version of this chapter has been published in the conference proceedings of PERVASIVE 2008.}
logged data and to self-assess the accuracy of it. While users were able to comment on the logged entries\(^2\) any time during the day, two additional self-assessment questions were required to be done at the end of the day. The experiment conductor would then review each participant’s feedback to ensure reasonable responses.

### Preliminary Lab Trial

There was also a week-long lab trial with five users. The aim was to improve the usability in order to facilitate accurate recording of participants’ physical locations inside the building. Modifications were made to the LocLog system for the actual logging phase:

- displaying the most recent user-logged and sensor-detected locations on the appropriate maps and in text (see Figure 5.3 on page 72);
- compiling a list of the users’ purposes for going to different places, allowing them to select an appropriate purpose for departing and arriving at a location;
- inserting the FVP (Frequently Visited Places) page;
- implementing the one-click box;
- detecting network instability and queueing the entries.

The LocLog Feedback system was also refined:

- adding a simple calculator to help estimate the accuracy of the entries;
- automatically highlighting problematic user entries;
- automatically highlighting and scrolling to relevant sensor readings when the user clicks on a logged entry.

### Participants

For the actual experiment, there were eight participants, including the author: seven males and one female. Their ages were between 22 to 35. There were one undergraduate and seven postgraduate students with Computer Science related majors. All of them regularly worked inside the building. Six of them were working in the same wing (and

\(^2\)The word *entry* is used throughout this chapter to refer to a user-logged message, which consists of a location name, an action, a timestamp of the entry and an optional message for the purpose of the action. A typical entry would look like: [Room 324, arrival, meeting/seminar, Tue Apr 3 18:28:15 2007], which denotes that a user arrived at Room 324 for a meeting or a seminar at 18:28:15 on 3 April 2007.
same level) of the building, another was in the other wing of the same level, and the
other one was on a different level. Seven of them worked at a personal space, somewhat
similar to cubicles, in open areas and one worked in an office. Figure 5.1 shows two
photographs with offices (left photograph) and open-area personal work spaces (right
photograph) in the SIT building.

Location Logging

We now describe the logging procedure in some detail because its validity is critical
to the remainder of the evaluation. The procedure for logging each entry normally
consisted of the following steps:

1. The participant selects the tab\(^3\) of the level in the building where they are currently
   located.

2. Then the participant clicks on the location they want to log on the map.

3. A message pops up just below the cursor asking for the participant’s intended
   action, which is one of: arrival, departure or stay (see Figure 5.3 on page 72). The
   primary actions are arrival and departure; stay would only be used if one
   forgets to indicate their arrival on time, i.e. delayed arrival. The participant can
   optionally indicate their purpose for that action from a drop-down menu. This en-
   ables the participant to add semantics to the data, which facilitates understanding
   of the log at the end of the day.

\(^3\)As defined in the Wikipedia, a tab in graphical user interfaces “is a navigational widget for switching

Figure 5.1: Photos of offices (left) and open-area personal work spaces (right) in the SIT building
4. After a location, an action and an optional purpose for the action are selected, a message box would pop up with the information to be sent to the PersonisAM server. The entry is delivered automatically after eight seconds, unless the participant decides to retract the action, by clicking **Undo** (see Figure 5.4 on page 73).

Each participant was asked to log their locations whenever they moved from one place to another—or when they realised they had forgotten to log an entry—inside the building. There were some exceptions: going to the toilet, short trips that normally took less than two minutes (e.g. getting water) and in transit from one place to another. For the first two exceptions, participants were asked to put in a **departure** and an **arrival** entries of the location they departed from. So, for example, when Bob wants to get some water in the nearby drinking fountain, he may log his departure from his office for getting a drink and his arrival to his office after he comes back.

### 5.2 System Design

Two systems have been designed for this study: the Location Log system, which allows users to record their movements inside the SIT building, and the LocLog Feedback system, which enables users to review, comment, and assess their logged locations.

#### 5.2.1 Location Log (LocLog)

LocLog has been designed for simplicity (i.e. intuitiveness and a gentle learning curve) and ubiquity. Since Tablet Personal Computers (Tablet PCs), shown in Figure 5.2, were the primary interacting interface, LocLog has also been developed to cater for

---

Figure 5.2: A Tablet PC was used for location logging inside the building
click-based interactions. LocLog allows users to log a location entry with three or less clicks:

1. selecting a floor level
2. selecting a place
3. choosing an intended action

As a stylus is the desirable input interface, tabs are used to allow larger viewing space, greater clicking areas, and to provide more flexibility for incorporating additional information, such as inserting an additional map.

Figure 5.3 is a screenshot of LocLog after user niu clicks on Room 324 on the map at the Level 3 tab. The system consults niu’s user model and discovers that Bob is a colleague of niu, so it displays not only the room number, but a personalised reference to the room both at the upper-centre region and in the pop-up message box.

**System feedback**

Feedback is an important means to engage users and prevent confusion, especially with a non-mouse/keyboard interacting interface. In a seemingly simple application like logging one’s own locations, there is still potential for misunderstandings, such as “Did I log when I arrived here?”, “Which room is this on the map?” Therefore, it was critical to exercise great care into giving appropriate system feedback. Feedback was provided in the following situations:

- When the users move the cursor over a building map, the area under the cursor would change colour and the location label would display the name of the place.

![Figure 5.3: A screenshot of LocLog when a user niu, who knows Bob, clicks on Room 324 on the map at the “Level 3” tab.](image-url)
5.2. SYSTEM DESIGN

This is illustrated at the centre upper area in Figure 5.3 where Bob’s Office (324) is. This reassures the users that they are pointing at the intended place.

- Floating/pop-up messages, such as the one in Figure 5.3, would fade away before they completely disappear from the screen. This continuous feedback prevents possible confusion from sudden disappearance of the messages.

- The most recent user-logged and sensor-detected locations are colour-coded and displayed both in text and as dots on the map. This, again, is illustrated at the bottom-left (i.e. [18:27:24] Level 3 West) and top-left corners (i.e. [12:13:26] arrival, Desk 3W32), as well as the two dots at the left central area in Figure 5.3 pointed out by the arrows. This addresses a common issue raised from the users: “Did I log when I arrived here?”, which may result in double entries or, worse yet, missing entries.

- The progress of processing an entry is communicated to the user by two messages: immediately after sending a logged entry (see Figure 5.4) and a confirmation message after the entry has been recorded. In the case of interrupted entry delivery, a different message would be displayed, as illustrated at the bottom right of Figure 5.5.

System availability

LocLog was implemented with a Web front-end accessible via the Hypertext Transfer Protocol (HTTP) to facilitate location logging throughout the building. There was Wireless Local Area Network coverage for most part of the building, so the participants could access the system through a Web browser with a Tablet PC. LocLog also accounted
for unstable network connections: it would store locations logged—with the original logging time preserved—during the period of network inactivity and automatically sent them when the connection resumed (see Figure 5.5).

Because of the rich AJAX (Asynchronous JavaScript and XML) interface used in LocLog, the saved entries would only be lost if participants closed the browser or shut down the machine.

Firefox was used as the browser for development, mainly because of its native support for Scalable Vector Graphics (SVG) and its popularity (i.e. so that users would be familiar with the interface).

### Accounting for human error

In addition to the three to five days of practice during the pre-experimental period, system design decisions have been made to minimise potential user mistakes:

- After the user sends away a logged entry, they still have a chance to retract that entry before the feedback message fades away: this accounts for the very plausible case that the participant accidentally clicks on the wrong part of the map, or clicks it at the wrong time.

- As there are usually a limited number of locations that a user would visit inside the building, a page of FVP is built that allows the user to log one of their top 10 most frequently visited places without having to pinpoint it on a map.

- A one-click entry box for departure is displayed after an arrival or stay entry has been submitted, following the assumption that a user would depart the place they last arrived at. Figure 5.5 shows an example of such a message: Departing Room 324.
5.2.2 Location Log Feedback

Participants were asked to review their log at the end of each day using the Location Log Feedback system (Figure 5.6) to identify inaccurately logged location data. The feedback system uses an online interface that consists of the following:

- logged entries for the day
- a list of location sensor messages
- two multiple choice questions on how informative the logged data was for that day
- input areas for users to give comments

The two multiple choice questions are for participants to assess the completeness and accuracy of their own logged entries (see (i) in Figure 5.6), and are to be completed after the participant has left the building for the day. The first question asks them to assess the reliability of the data they logged and the second one asks about the reliability of the logged data after applying corrections they provided. So, for example the participant indicated in question one that with the logged data, other people can usually find them, and this corresponds to roughly more than 75% of accuracy. This means, with the logged data, the user can be accurately located more than 75% of the time.

Desirable requirements for this feedback page, as well as how they have been addressed are listed below.

**Facilitate immediate error reporting.** This system is designed to enable users to comment on any problem in location logging, such as interrupted network or missed entries. The comments are saved every time the participant submits the data and are reloaded at each revisit. Participants were therefore asked to give feedback via the system whenever an event that could affect the accuracy of the logged data occurred.

**Help participants identify inaccurate and missing entries.** Potentially problematic entries, i.e. not in the sequence of arriving PlaceX and departing PlaceX, (categorisation of user logged entries are to be discussed in Section 5.4.2 on page 80) are highlighted in red to catch the participant’s attention (see Rows 6 and 9 in Figure 5.6). When a logged entry is clicked, the rows in the Sensor table that are within 2 minutes and 20 minutes to the corresponding entry are highlighted in green and lighter green respectively. This allows the participant to use relevant sensor data to identify errors in the entries. For example, when Row 8 in the
Figure 5.6: The Location Log Feedback system. The table to the left, the LocLog table, shows user-logged entries: (a) is the ID column that numbers each entry; (b) shows time of each logged entry as well as the gap between this one and its preceding entry; (c) shows the action and purpose, if given, of the entry; (d) is the location logged; (e) is for users to give comments for each entry; (f) is the duration for the day; (g) shows the two highlighted rows that are potentially erroneous; (h) is a basic time calculator for users to better estimate the quality of the logged entries; (i) contains the two self-assessment questions to be completed at the end of the day; and (j) is the area where the user puts more comments about the entries. The table to the right, the Sensor table, lists the evidence sources from sensors: (1) is the ID column; (2) is the timestamp of the evidence source; (3) is the sensor message; (4) lists the inferred location as well as the sensor type (i.e. BT for Bluetooth sensors, Sys for activity sensors and Login for login sensors), and (5) shows the highlighted rows, temporally close to the 8th logged entry clicked by the user; it highlights the rows that are less than 2 minutes away (i.e. 2 minutes backward and forward) from the clicked entry with a darker green colour, and those that are less than 20 minutes away with a lighter green colour.
LocLog table in Figure 5.6 is clicked, the temporally relevant rows in the Sensor table are highlighted and scrolled to. These give evidence that the participant was in the building again around 16:09 (Rows 86–87 in the Sensor table), about one hour earlier than the logged arrival at 17:07 (Row 9 of the LocLog table). Hence, there might be missing entries between Rows 8 and 9 of the LocLog table. This feature of automatic highlighting and scrolling is especially helpful when a large number of sensor readings—typically about 100–300 readings per day per user—are reviewed.

**Make self-assessment more objective.** Textual descriptions for the answers of multiple choice questions (e.g. *other people can almost always find you*) convey the meaning of each answer to each participant in a more intuitive way. However, with users coming from different backgrounds, non-quantitative answers can lead to biased and subjective results. Therefore, each choice of the questions is given a corresponding percentage of accuracy for a more discrete reference point. To help the participant better approximate the percentage of (un)reliable time, the system displays the amount of time that the participant stays in the building for the day (shown in the LocLog table header in Figure 5.6 as 576 minutes) as well as the duration between each pair of consecutive entries (the number between parentheses after each time value). The total time is calculated by subtracting the time of the last entry from that of the first entry. This may inevitably include the time when the participant was actually outside of the building during the period, which should be excluded. Rows 8 and 9 in Figure 5.6 illustrate this issue: the participant spent most of the time between 12:27 and 17:07 outside of the building and that should be subtracted from the total time shown. The participants were advised to account for this issue when calculating the amount of unreliable time to answer the self-assessment questions. In addition, a basic time calculator is available just under the left table for participants to better estimate the accuracy (see (h) in Figure 5.6).

### 5.3 Sensor Infrastructure

A representative range of sensor types has been deployed for this study, taking account of pragmatic considerations, such as minimal maintenance, simple setup and minimal user interaction. There were three types of sensors used for this study: Bluetooth sensors, activity sensors, and login sensors. They have been described in Section 4.3.3 on page 58. For this study, there were a total of 16 Bluetooth sensors, 16 Bluetooth-enabled devices
(5 mobile phones, 8 Tablet PCs, and 3 laptop computers), 12 activity sensors, and the login sensors. Figure 5.7 shows the Bluetooth sensors, as white circles, deployed in the SIT building.

This setup suits especially well—with technology replacing much physical contact (e.g. emails, instant messaging, online multimedia communication)—in an office environment where people tend to spend a significantly longer period of time at their office space (González and Mark 2004). Regarding the reliability of the sensors, ambiguous location information generated from different sensors is very common. For example, when Bob, who has a Bluetooth-enabled mobile phone, is sitting in front of a machine with an activity sensor installed and is also within the range of a Bluetooth sensor, both sensors would infer the presence of Bob.

5.4 Analysis on User Data and Sensor Data

This section reports the data collected from the study as well as discusses its implications. These include analysis on participants’ feedback and self-assessment on the entries, user-logged data, sensor data, and location resolution with the established control.

5.4.1 Participant Feedback and Self-assessment on LocLog Entries

Participants used the feedback box mostly to report erroneous entries and omissions with corresponding corrections. Whenever possible, these corrections were manually put into the logged data by the experiment conductor. Other feedback included network instability, Tablet PC out of battery, user interface suggestions and delayed logging for

![Figure 5.7: Sensors used in the SIT building for the LocLog study: Bluetooth sensors are shown as white circles, activity sensors are shown as black circles, login sensors are actively scanning on computers in the Level 1 labs shown as green squares.](image)
User self-assessment was the two multiple choice questions regarding reliability of the user-logged data, as described in Section 5.2.2 on page 75. The exact questions are shown as (i) in Figure 5.6 on page 76. Table 5.1 summarises the participants’ answers to the two questions. The columns labelled from 1 to 5 indicate the total number of times each answer was given by each participant. Rows marked with an uppercase letter from A to H represent the eight participants. Participant A is the author who conducted this experiment. For example, participant A answered 1—meaning “(almost) always informative” or “roughly 90%+ accurate”—23 times, answered 2 nine times, and answered 3 and 4 once each to Question One, which added up to 34 times, or days for the whole experiment.

The last row summarises the self-assessment statistics for all participants. Similar to the last row, the second to the last row summarises the data, but only for participants B to H. This row was added because Participant A undertook a significant longer period of time for the study than the others, and that could bias the result. The low averaged numbers—1.48/1.46 and 1.13/1.14 for Question One and Question Two respectively—indicate that participants were generally confident of the accuracy of the location information they provided, especially after reviewing their own data and indicating any omissions and errors at the end of each day. There were six days rated as 2, three days rated as 3 and one day rated as 4 in Question Two: those days of data have been omitted from the later evaluation process, for better data integrity.

Table 5.1: A summary on reliability questions answered by each participants. Answer are from 1–5, ranging from most reliable (roughly 90%+ accuracy) and least reliable (roughly less than 25% accuracy).

<table>
<thead>
<tr>
<th>User</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
<th>Mean</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>34</td>
<td>1.41</td>
<td>32</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>34</td>
<td>1.15</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>1.60</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>1.30</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.36</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.00</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>1.67</td>
<td>9</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>1.33</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.73</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.18</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.55</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>1.09</td>
</tr>
<tr>
<td>G</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>1.38</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>1.00</td>
</tr>
<tr>
<td>H</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>1.00</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1.00</td>
</tr>
<tr>
<td>B–H</td>
<td>47</td>
<td>23</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>77</td>
<td>1.48</td>
<td>69</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>77</td>
<td>1.13</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>32</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>111</td>
<td>1.46</td>
<td>101</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>111</td>
<td>1.14</td>
</tr>
</tbody>
</table>
5.4.2 User-logged Data

A more objective measure we took was to examine the statistics of the actual logged data. Table 5.2 shows a summary of statistical information on user-logged data. Rows headed with uppercase letters present the data of the eight participants. Columns represent, respectively, the total number of days a user had logged their locations, total number of unique locations logged, minimum and maximum number of entries in a day, total number of entries throughout the experiment and an average number of entries per day. So, for example, participant A provided 34 days of logged data during the analysis phase, with 45 distinct locations logged. The minimum number of entries in a day was 11 and the maximum was 52. There was a total of 882 entries for the 34 days and an average of 25.9 entries per day for that participant.

In addition to the overview of the data given by Table 5.2, each pair of consecutive logged entries was also analysed. There are three possible actions: departure, arrival and stay. When these are combined with a binary location value—same location or different—there are 18 different combinations. One example of the combination is “departing Room 123” and “arriving Room 320”. We categorise the 18 combinations into four different classes of time: stationary, transit, unknown and error. Table 5.3 lists the how often each combination of actions occurred during the experiment, what behaviour was implied, and how many logging errors are implied by each combination. We now discuss each of the four classes of time with references to Table 5.3.

**Stationary time.** It denotes the amount of time that the participant was located at a

<table>
<thead>
<tr>
<th>User</th>
<th>Days</th>
<th>Unique Locations</th>
<th>Number of Data Entries</th>
<th>Total</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min. (per day)</td>
<td>Max. (per day)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>34</td>
<td>45</td>
<td>11</td>
<td>52</td>
<td>882</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>12</td>
<td>82</td>
</tr>
<tr>
<td>C</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>23</td>
<td>130</td>
</tr>
<tr>
<td>D</td>
<td>12</td>
<td>10</td>
<td>2</td>
<td>15</td>
<td>103</td>
</tr>
<tr>
<td>E</td>
<td>11</td>
<td>5</td>
<td>8</td>
<td>18</td>
<td>114</td>
</tr>
<tr>
<td>F</td>
<td>11</td>
<td>20</td>
<td>6</td>
<td>36</td>
<td>217</td>
</tr>
<tr>
<td>G</td>
<td>13</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>76</td>
</tr>
<tr>
<td>H</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>64</td>
</tr>
<tr>
<td>B–H</td>
<td>77</td>
<td>38</td>
<td>2</td>
<td>36</td>
<td>786</td>
</tr>
<tr>
<td>All</td>
<td>111</td>
<td>61</td>
<td>2</td>
<td>52</td>
<td>1668</td>
</tr>
</tbody>
</table>

Table 5.2: A summary of user-logged data
Table 5.3: All possible combinations for any two consecutive user-logged messages

<table>
<thead>
<tr>
<th>Combination</th>
<th>Occurrences</th>
<th>Implied Behaviour</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 arvl X → dptr X</td>
<td>778 (50.0%)</td>
<td>Stationary, i.e. at the place X</td>
<td>0</td>
</tr>
<tr>
<td>2 stay X → dptr X</td>
<td>22 (1.4%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 dptr X → arvl Y</td>
<td>496 (31.9%)</td>
<td>Travelling from place X to place Y</td>
<td></td>
</tr>
<tr>
<td>4 dptr X → arvl X</td>
<td>186 (11.9%)</td>
<td>Outside of the SIT building, a short trip or missing both arrival and departure</td>
<td>0 or 2</td>
</tr>
<tr>
<td>5 arvl X → arvl Y</td>
<td>16 (1.0%)</td>
<td>Missing a departure entry from place X</td>
<td>1</td>
</tr>
<tr>
<td>6 stay X → arvl Y</td>
<td>2 (0.1%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 dptr X → dptr Y</td>
<td>10 (0.6%)</td>
<td>Missing an arrival entry for place Y</td>
<td>1</td>
</tr>
<tr>
<td>8 dptr X → stay Y</td>
<td>11 (0.7%)</td>
<td>Delaying an arrival entry for place Y</td>
<td>1</td>
</tr>
<tr>
<td>9 dptr X → dptr X</td>
<td>12 (0.8%)</td>
<td>Row 4 + missing an arrival entry for place X afterward</td>
<td>1 or 3</td>
</tr>
<tr>
<td>10 arvl X → arvl X</td>
<td>9 (0.6%)</td>
<td>Row 4 + missing a departure entry from place X beforehand</td>
<td>1 or 3</td>
</tr>
<tr>
<td>11 stay X → arvl X</td>
<td>1 (0.1%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 dptr X → stay X</td>
<td>5 (0.3%)</td>
<td>Row 4 + delaying an arrival entry for place X afterward</td>
<td>1 or 3</td>
</tr>
<tr>
<td>13 arvl X → stay Y</td>
<td>0</td>
<td>Missing a departure entry from place X and delaying an arrival entry for place Y</td>
<td>2</td>
</tr>
<tr>
<td>14 stay X → stay Y</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 arvl X → dptr Y</td>
<td>2 (0.1%)</td>
<td>Missing both a departure entry from place X and an arrival entry for place Y</td>
<td>2</td>
</tr>
<tr>
<td>16 stay X → dptr Y</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 arvl X → stay X</td>
<td>2 (0.1%)</td>
<td>Row 10 + delaying an arrival entry for place X</td>
<td>2 or 4</td>
</tr>
<tr>
<td>18 stay X → stay X</td>
<td>5 (0.3%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1557 (100%)

place. This category includes two combinations of entries: arrival X→departure X and stay X→departure X, where X is a place. These are represented by the first two rows in Table 5.3. One example would be “arriving Room 100” and “departing Room 100”, meaning the participant logged arrival at Room 100 and the next logged entry was departing Room. As indicated in Section 5.1 on page 68, the stay action indicates a delayed entry for arrival. Even though there is uncertainty about the time of arrival, the time between stay and departure for a location still indicates the person being at that location. This is the only category
of time taken into account for accuracy calculation.

**Transit time.** This is the amount of time that the participant was travelling from one place to another. It is represented by departure \( X \rightarrow \text{arrival} \ Y \), or Row 3 in Table 5.3. One example would be “departing Room 100” and “arriving Room 300”. This category of time is excluded from analysis for two pragmatic reasons: the sensor infrastructure does not (intend to) provide the precision of walking movements; and it accounts for a relatively small proportion of time (around 3.5%, see Figure 5.8(b)) which outweighs the high uncertainty.

**Unknown time.** It denotes the amount of time that the location of the participant could not be inferred from the entries. This category is represented by departure \( X \rightarrow \text{arrival} \ X \), or Row 4 in Table 5.3. An example would be “departing Room 100” and “arriving Room 100”. Possible explanations for this type of entry are: participants went out of the building; they took short trips around their work space; or they failed to log both arrival and departure of another place. Manual inspection of the data revealed that 90% of the cases were likely to be among the first two explanations. That leaves the rest of the 10% to be potentially erroneous entries. This class of time is also excluded from analysis as the actual location of the participant is unknown.

**Error time.** This is the amount of time when the participant made one or more errors in logging an entry. This category of time consists of all 14 remaining combinations, described in Row 5–18 in Table 5.3. One example combination is departure \( X \rightarrow \text{departure} \ Y \). This implies that the user failed to put in an arrival entry when arriving the place Y. Those can be further divided into three basic errors and four compound errors. The basic errors are: missing a departure entry (Row 5 and 6), missing an arrival entry (Row 7) and delaying an arrival entry (Row 8). Compound errors are essentially combinations of basic errors. This category accounts for about 4.7% of overall entries and the majority (2.5%) of them imply the basic errors, i.e. one human error. It indicates that participants were generally careful when putting in each entry.

Figure 5.8(a) and Figure 5.8(b) summarise the distribution of the logged data in terms of the numbers of entries and amount of time, respectively, for each class. In sum, stationary time accounts for 51% of total number of entries and 78.5% of the time, which is about 6.5 hours from 9am–5pm, working hours. Most of the unknown time, which accounts for about 15.1% of the time, was because of participants going out of the
5.4. ANALYSIS ON USER DATA AND SENSOR DATA

(a) Number of entries

(b) Amount of time: logged entries (left) and sensor (right)

Figure 5.8: Classifications of the participants’ behaviour in terms of entries (a) and time (b)

building. A relatively small amount of time was classified as error (2.9%) and transit (3.5%).

Discussion about Individuals

Figure 5.9 illustrates the amount of time each participant’s logged entries imply in each of the time classes defined earlier on page 80: stationary, transit, unknown, and error. The x axis represents each participant, an average between participants B to H, and an average of all participants, and the y axis denotes the distribution of each time class in percentage. The bottom portion of each bar represents the length of time a participant had explicitly indicated their location (i.e. stationary time). The next portion in the stack indicates the length of time a participant is in transit from one place to another within the building. The next one up is the amount of unknown time, and the top portion in the stack is the amount of time that appears to be logged erroneously.

As the experiment conductor, Participant A had to ensure the experiment ran smoothly that required him to be highly mobile (e.g. talk to other participants, test sensors), hence a relatively larger number of entries logged: 25.9 entries per day, compared to an average of 10.2 entries per day for other participants (see Table 5.2). Participant B often had lunch at their work space, which contributed to less unknown time and more stationary time (see Figure 5.9). Participants D, G, and H were less mobile than the others, and they were not coffee drinkers, so they did not travel down to the coffee room often; this resulted on little amounts of transit time and lower-than-average entries per day. Participant E spent some time in a laboratory of the building that had limited
converge of Bluetooth sensors and that gave a significantly larger amount of unknown time. Participant F made several trips to coffee room each day and tended to spent time talking to people they saw on the way, and that consequentially increased the transit time. They also attended lectures outside of the building, which explained the larger percentage of unknown time. Participant G had a minimal percentage of error time, mainly because they only logged two unique locations: 97% of stationary time was spent at their work space and the other 3% was spent at a seminar room.

5.4.3 Sensor Data

A diverse range of sensor and device evidence was gathered during the study: 38414 messages from the 16 Bluetooth sensors scanning 16 Bluetooth-enabled devices, 13252 messages from the 12 activity sensors and 57 messages from the login sensors. Appendix C on page 137 displays a table with an overview of the more detailed statistics of the device and sensor messages.

We classify the evidence into three categories: no evidence, unanimity and conflicting evidence. No evidence denotes the amount of time that had no sensor evidence inferring the participant’s location. Time classified as Unanimity means that when there is a
uniform location value from the sensor evidence, i.e. no ambiguous evidence sources. *Granularity variance*, described in Section 1.2 on page 9, represents the amount of time when the ambiguous evidence sources are overlapping in granularity difference. *Others* denotes the percentage of time with all other types of conflicting evidence sources; from manual inspection, more than 99% of the actual time was *false positives*, according to the classification in Section 1.2 on page 9. Table 5.4 summarises the classes of sensor evidence collected during the study in percentage of time.

**Discussion about Individuals**

The high percentage of granularity variance (57.0%) was because each participant was detected by at least one Bluetooth sensor and one activity sensor when they were using their computer at their work spaces. Participant C’s and D’s data have more unanimous evidence than the other three. This is because their Tablet PCs would enter stand-by mode more often, causing their system sensors to be the sole sensing devices. Participant E’s higher percentage of *No evidence* time (4.1%) was mainly due to their time spent in a research laboratory outside of the Bluetooth sensor coverage. Because the work spaces for participants A, B, C, F and H could be detected by Bluetooth sensors from different floors of the building—especially for participant H—they had more non-granularity-variance conflicting evidence than Participants D and E. Participant G changed locations least frequently throughout the study; they only travelled to two

---

4Each Tablet PC would enter stand-by mode after 30 minutes of inactivity on battery.

Table 5.4: Classification of sensor evidence in percentage of time

<table>
<thead>
<tr>
<th>Participant</th>
<th>No evidence</th>
<th>Unanimity</th>
<th>Granularity variance</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1%</td>
<td>14.4%</td>
<td>71.2%</td>
<td>14.2%</td>
</tr>
<tr>
<td>B</td>
<td>0.0%</td>
<td>17.6%</td>
<td>71.6%</td>
<td>10.7%</td>
</tr>
<tr>
<td>C</td>
<td>0.3%</td>
<td>51.6%</td>
<td>32.2%</td>
<td>15.9%</td>
</tr>
<tr>
<td>D</td>
<td>0.0%</td>
<td>42.6%</td>
<td>48.3%</td>
<td>9.0%</td>
</tr>
<tr>
<td>E</td>
<td>4.1%</td>
<td>28.5%</td>
<td>65.0%</td>
<td>2.4%</td>
</tr>
<tr>
<td>F</td>
<td>0.2%</td>
<td>21.9%</td>
<td>59.0%</td>
<td>18.9%</td>
</tr>
<tr>
<td>G</td>
<td>0.0%</td>
<td>26.0%</td>
<td>73.8%</td>
<td>0.2%</td>
</tr>
<tr>
<td>H</td>
<td>0.0%</td>
<td>0.2%</td>
<td>0.0%</td>
<td>99.8%</td>
</tr>
<tr>
<td>B–H</td>
<td>0.5%</td>
<td>29.3%</td>
<td>50.4%</td>
<td>19.7%</td>
</tr>
<tr>
<td>Average</td>
<td>0.4%</td>
<td>24.6%</td>
<td>57.0%</td>
<td>18.0%</td>
</tr>
</tbody>
</table>
places during the study: their work space (97%) and a seminar room (3%).

Participants A, B, D, F, and H all had more than one Bluetooth-enabled devices, even though participant B’s phone was only detected intermittently due to its unstable Bluetooth signal. This provided more evidence for their current locations, but could also generate more conflicting information (e.g. when they left the mobile phone at their desks when going to a meeting). Participant D only turned the Bluetooth of the laptop on discoverable mode\(^5\) occasionally. In this case, it was only turned on for a day when they were in the lecture theatre. Participants B, D, E, and H used the same laptop at both university and home; so their activity sensors on the laptops tended to have proportionally more evidence than the other four participants. In summary, each participant had evidence sources from at least nine sensors, at most eighteen sensors; all had more than 1,800 pieces of evidence from those sensors.

Figure 5.10 shows roughly four and a half hours of both logged and sensor-generated location data. The x axis indicates the time and the y axis indicates places in the

\(^5\)This is the mode when a Bluetooth device can be seen and discovered by other Bluetooth devices, such as a Bluetooth sensor.
building. Asterisks denote user-logged data and green dots represent sensor evidence. Horizontal lines connecting asterisks denote time when the participant was stationary in one place (i.e. the first point denotes arrival and the next one denotes departure) and diagonal lines could mean either that a person is in transit from one place to another (i.e. the first point denotes departure and the next one denotes arrival) or some form of error had occurred (usually omissions). So, in Figure 5.10, participant E arrived at the office from home at about 2:30pm, and then went to Room 203 and Room 125 for some time before he departed from office around 6pm. The participant put in a total of 12 logged entries for that day. As they were staying at their office, two sensors were constantly detecting them: activity sensor on their computer and a Bluetooth sensor installed at Level 3 East. When they went to Room 203, a Bluetooth sensor on Level 2 West sensed their presence throughout their stay. Later when they went down to Room 125, a meeting room, two sensors along the way—on Level 1 Middle and Level 1 East—sensed their arrival and departure, but not their stay, because the Bluetooth signal was simply not strong enough in the meeting room.

Figure 5.11 illustrates both user-logged and sensor data for an active and a less active participants in the study. The graphs span the same period of time, 16 days including weekends. There are twice as many locations in the Figure 5.11(a) than Figure 5.11(b), 36 and 18 respectively.

Figure 5.11: An overview of both logged and sensor data for the most and the second least mobile participants
5.4.4 Resolving Location Values

In this section we compare accuracy using ontological algorithms with three representative, non-ontological algorithms, as described in Section 3.4.1 on page 39. The algorithms used in the analysis include the following:

- **Ontological algorithms:**
  - Granularity Harmoniser (GH)
  - Closest Common Subsumer (CCS)
  - Democratic with GH (D&GH)

- **Non-ontological algorithms:**
  - Point
  - Time Decay (TD)
  - Bias

- **Best Possible**

We compare the accuracy on resolving location values across different levels of location granularity. Four different grain-sizes of locations are used: Room/Desk (e.g. Room 300, Desk 3W30), MultiRoomUnit (e.g. Level 3 West), LevelInABuilding (e.g. Level 3) and Building. Only the results of the first two grain-sizes are presented, as they represent the granularity of locations detected by the sensors used in this study.

It is sometimes not possible to resolve the location to the desired grain-size with the existing sensors, and this may be caused by system faults or lack of sensor coverage. When a user, for example, is not actively using a machine that either runs an activity sensor or is monitored by a login sensor, the finest-grained location value for that user is not likely to be generated. Thus, a resolver function which estimates the best possible location value that can be achieved is also used. A *best possible* location is defined as the best matched location value—in terms of granularity difference from the user-logged location—from the list of available sensor evidence sources. So, for example, if the user-logged location is Room 100 and the list of possible locations from sensor evidence are Level 1 and Level 3, Level 1 would be the best matched location value. For all cases, when there is no evidence found in the designated time frame, the last resolved location value would be used.
5.4. ANALYSIS ON USER DATA AND SENSOR DATA

When comparing the resolved location values from sensors against the user-logged location value, both location values are normalised to the desired grain-size whenever possible. For example, if the user-logged location is Room 300, and the resolved value from sensors is Level 3, it would be counted as an erroneous sensor value for Room/Desk and MultiRoomUnit grain-sizes, as Level 3 can not be normalised to a finer-grained location. However, it would be correct for LevelInABuilding and Building grain-sizes, as both Room 300 and Level 3 would be normalised to Level 3 and the Building respectively.

Location Reasoning at Room/Desk Grain-size

Figure 5.12 shows the error rates for running five algorithms and the Best Possible resolver on the participants’ data to resolve for the location at the Room/Desk level, in the order of the mean error rates. The x axis denotes the eight participants and their average. The y axis denotes the accuracy of each resolver with respect to the participant’s data measured in percentage of errors. For example, for participant A, the Point algorithm is able to resolve a location value for that participant when they were in the building with an error rate of 67.5%, the Democratic and Granularity Harmoniser (D&GH) algorithm (see Algorithm 1 on page 44) reduces it to 20.3% and the Best Possible is 19.2%. The results of CCS (Closest Common Subsumer) are excluded from the chart for its poor performance, which would reduce the visual clarity of the chart. It constantly performed more than 40% worse than the Point algorithm.

![Figure 5.12: Error rates for different resolvers reasoning at Room/Desk grain-size](image-url)
The ontological algorithms—GH and D&GH—generally have lower error rates than Point and Time Decay (TD). Participant H is an exception, because their work space constantly received inter-floor Bluetooth signals *(false positive evidence)* which resulted in extremely high non-granularity-variance conflict (99.8%). On average, GH reduces the error rates of Point and TD from 54.5% and 45.2% to 24.9% by harmonising granularity variance in sensor evidence. The D&GH algorithm filters out some false positive noise before resolving granularity variance, hence further reduces the error rates. The Bias algorithm performs comparably to D&GH because it favours the participants’ work spaces, where they spent most of the time.

**Location Reasoning at MultiRoomUnit Grain-size**

Figure 5.13, analogous to Figure 5.12, illustrates the error rates generated by a list of resolver functions in reasoning about location at the MultiRoomUnit grain-size. CCS is, again, excluded. Note the high error rates on Point (44.9%), TD (24.2%) and GH (44.9%) algorithms for participant H.

The Point and TD algorithms now have comparable error rates. GH performs the same as Point, because GH only filters out granularity variance before using the Point algorithm for the other conflicting evidence. There are currently two granularity levels
in our sensor infrastructure: Room/Desk and MultiRoomUnit, so GH would only differentiate itself from Point when reasoning at the finer grain-size. D&GH has slightly better mean error rates than Bias at this grain-size, mainly due to its ability to filter out false positive noise. For instance, it is able to filter out the evidence value of Level 2 East from the possible locations of Level 1 East, Level 1 Middle and Level 2 East, which can not be done with the other algorithms. The location values resolved by CCS, in most cases, are too coarse-grained to be useful under this sensor infrastructure.

Participant E and F spent more time away from their computers, usually in a laboratory, the common room, or areas nearby their work spaces. This often prevented the sensors from getting their finest grain locations, which resulted in higher error rates for their best possible results. As a reminder, the best possible algorithm selects the best matched location value from the list of available sensor evidence sources. Bias and D&GH show clear improvements over the other algorithms for participants A, B, C and H because they have more non-granularity-variance conflicting evidence than participants D, E and G.

5.5 Summary

The key contribution of this chapter is the exploration of a new role for ontological reasoning in pervasive computing: we have tackled one of the foundation problems, modelling location from a diverse range of sensors, each of which produces evidence at varying levels of noise and uncertainty. We have also presented an experimental procedure for obtaining reliable records of people’s movements inside of a building.

We have described how, in practice, the implemented framework perform in interpreting the location evidence. This shows that ONCOR is able to resolve location values at different levels of granularity with a semi-automatically generated working location ontology. Our experimental evaluation compared the performance of these algorithms against a gold standard, created by participants in our study.

From the experimental results, GH reduced the error rates of the Point algorithm by over one half on average, from 54.5% to 24.9%, demonstrating the the power of an ontology in resolving granularity variance. D&GH further reduces the mean error rate of GH from 24.9% to 16.1%, showing the ability of the ontology to resolve false positive sensor evidence. In terms of the other type of conflict, false negative, it is yet to be explored.

Each algorithm may perform well, in terms of both accuracy and precision, in different contexts, as discussed in Section 3.4.1 on page 39.
Even in cases where ontological reasoning does not give minimal errors, it can provide more generic, less context-dependent reasoning in a dynamic environment. By contrast, the Bias algorithm can only work well for people who spend more time at their work space. With an approach like ours, based on ontological reasoning, it becomes possible to reason across granularity levels. There has been considerable work on ontological reasoning and modelling for pervasive environments, as described in Chapter 2, reflecting the intuitive appeal of such approaches. Our work goes beyond the earlier proposals to the implementation and evaluation of location reasoning across granularity levels to conclude a person’s location, exploring and comparing different ontological algorithms. Our empirical study provides insights into the ways these algorithms operate for different people with different behaviours.

Ontological reasoning appears to offer much promise for pervasive computing because it seems to provide a way to deal with key problems, including reasoning about a person’s location, based on a diverse set of evidence from different classes of sensors.
Chapter 6

Pervasive Personalisation Reasoning

In this chapter, we evaluate PECO’s power in the critical area of personalisation in pervasive computing. There are three aspects to this evaluation process.

Proof of concept. We implemented PERSONAF by augmenting an earlier version of a location modelling system (Assad et al. 2007) by adding the PECO and ONCOR. The Adaptive Locator demonstrates that these parts can support the creation of applications that make use of the PERSONAF framework.

Ontology-evaluation. We conducted an evaluation based upon an application that makes use of the PECO ontology and the associated ONCOR reasoning, which constitutes an application-based ontology evaluation (Brank et al. 2005).

User evaluation. A user study was conducted to show the effectiveness and usability of the personalised Adaptive Locator application.

In Adaptive Locator, we show how PERSONAF facilitates personalised reasoning, in particular, by displaying people of relevance and personally meaningful location labels. Relevance largely depends on the context and the user. For the purpose of the user study, we narrowed it down to mean the closeness of relationship between the user and people discovered by the system. Importantly, we also examine PERSONAF’s capability to provide understandable explanations of personalisation to end-users. So, for example, when Bob’s location is queried, a system like Adaptive Locator may use PECO and ONCOR to produce “Bob is at the social hub”. Should the user think the reasoning

\footnote{A shorter form of this chapter has been published in the conference proceedings of AH 2008.}
of “social hub” incorrect or incomprehensible—or if they are just curious about this description—the system can then provide an explanation, such as “According to the 2008 postgraduate student handbook, Room 125 is a social hub”.

6.1 System Overview

Adaptive Locator displays the location of people in our building by making use of PERSONAF framework to provide personalised and scrutable information. Figure 6.1 shows an anonymised snapshot of the Adaptive Locator interface for Level 3, or the third floor, of the smart building. The top left allows navigation to other floors: as there were no people detected on level 2, it was hidden at this time. Each dot on the map represents a person and its darkness indicates the inferred closeness of the relationship to the current user. So, for example there is a dark dot marked close colleague. The user can click “what do the colours mean?” to see this explained (see Figure 6.2).

Below the map is personalised information about the people on that floor. Holding the cursor over a person’s location description highlights the dot presenting that person.

---

2 Actual names have been substituted with fictional ones.

Figure 6.1: A screen shot of the Adaptive Locator as seen by the user marked current user
6.1. SYSTEM OVERVIEW

In the figure, the user is holding the cursor over at the place the hand appears, his desk. Clicking the textual description of a place, such as his desk or a person (e.g. Ian) gives an explanation of the personalisation, as we describe below. The system omits people who it infers to be unknown to the user, but the user can click “3 person(s) hidden” (at the bottom left of Figure 6.1) to display them.

The system uses the Social Connection Resolver, described in Section 3.4.2 on page 44, to reason about the user’s relationship with people to determine who to display and who to hide. The system models relationships by collecting evidence from available resources. For this evaluation, a person is assessed as closer if they work at a nearby office or desk and if they are on the same internal mailing aliases. More sophisticated reasoning could also be done: for example, hide people who are inferred to be less closely related to the user; allow the user to show or hide arbitrary groups of people, such as people sitting nearby and people working on the same project; or make use of people’s social connections through, for example, social networking websites, such as Facebook\(^3\), MySpace\(^4\) and LinkedIn\(^5\). In this study, we limited the number of sources, as the main focus is to explore PECO’s capability to deliver personalised and scrutable information and to evaluate whether this is indeed a plausible approach.

To personalise location labels (e.g. his desk) such as shown in Figure 6.1, a range of evidence is collected from PECO. Figure 6.3 illustrates the full explanation (i.e. each “why?” link is clicked to have the evidence source listed) of why Ian’s location is presented as his desk. The system makes the reasoning with the following inferences and evidence sources:

- Ian is currently at Desk 3W32. This is inferred from the sensor evidence of the location of System Sensor with id ian@pc-g61b-1, detected on Thu Jan 10 16:28:25 2008.

- Ian’s work space is Desk 3W32, so the system can map the place to their desk.

\(^3\)http://www.facebook.com/, 31 Dec 2008.
\(^4\)http://www.myspace.com, 31 Dec 2008
CHAPTER 6. PERVERSIVE PERSONALISATION REASONING

This inference is based on the evidence collected from the postgrad student directory on Thu Jan 10 16:28:25 2008.

- The user knows Ian, so “Ian’s Desk” is meaningful. This is inferred from two evidence sources: that Ian’s work space is adjacent to the current user, and they are both on the chai_list email alias.

- The user knows where Desk 3W32 is, which is currently inferred because the user has been to the wing containing that place. The Been-to Places Resolver, described in Section 3.4.2 on page 44, is used to find places, at the multi-room unit granularity, that the current user has been to in the last 90 days. Then the spatial subsumption relation is checked between Desk 3W32 against each wing that user has been to. For this user, evidence shows that they have been to Level 3 West on Thu Jan 17 17:05:35 2008, and Desk 3W32 is spatially subsumed by Level 3 West. Therefore it is inferred that the user may know where Desk 3W32 is.

6.2 The User Study

We conducted a within-subject user study comparing Adaptive Locator against a non-adaptive version (see Figure 6.6). The evaluation included the following key goals:

- assessing the accuracy and value of the selection of people relevant to the user;

---

6 Ideally we would want to gather evidence about whether the user has been to the place, but the current sensor infrastructure did not allow such information. This could be improved by deploying finer-grained physical sensors.
6.2. THE USER STUDY

- assessing the value of personalised labels;
- assessing the understandability of the explanation for personalisation.

We designed a set of tasks for both versions of the Locator system. Figure 6.4 shows the way that we used two monitors for the experiment. We used the left screen to display the tasks. Participants used the system with the right screen. For both systems, the user was asked to locate a particular person and to identify people in a certain area, in order. This assessed the usability of the adaptive and non-adaptive systems. The user was also asked to indicate their knowledge about a familiar colleague’s office location from the textual descriptions (e.g. Judy’s Office, Room 324) for both systems in order to assess the value of personalised location labels. Note that the nature of the experiment in the personalised condition means that different users worked with somewhat different displays.

Before the main task-sets, participants did four familiarisation tasks. This gave familiarity with the map interface. They then completed tasks for each of the two systems using a cross-over methodology to reduce inter-subject variability: half of the participants used the adaptive system first, and the other half used the non-adaptive one first. After the tasks, participants completed an online questionnaire with eight questions. Each asked for the level of agreement on a seven-point Likert scale, with space for comments and justifications of the answer. Participants were observed by the

Figure 6.4: The user progressively worked through the tasks on the left screen as they used the system on the right screen.
author throughout. The complete set of tasks are included in Appendix D on page 139.

6.2.1 Participants

Eight participants were recruited: two women and six men; two staff members, two graduate students, and four undergraduates. Their ages ranged between 19 to 57. Six worked regularly in the building for over six months, and the other two for four months and one month respectively. All but one had used an earlier non-personalised version of Locator (Assad et al. 2007), and the other person knew about it before the study. As we made use of inferences about social networks, we deliberately recruited the participant group as users of the building who are familiar with the maps. Figure 6.5 indicates the work space location of each participant. The black circles represent the four participants who did the tasks for the non-adaptive system first, while the white circles present the others who did the tasks for the adaptive system first.

6.2.2 Systems

To ensure the validity of the study, we carefully designed the systems to test the desirable variables: personalised selection of relevant people, personalised location labels, and explanation of personalisation. We took a time-snap of the system, so each participant would have the same set of information across both systems. This, however, means that the data was not delivered real-time and could have potentially caused some confusion for the participants. For example, the participant might find that User A was not actually sitting at his desk. The participants were explicitly informed about this time shift before commencing the system tasks.

Figure 6.5: A map-view of the work spaces for the participants
6.2. THE USER STUDY

The designed differences between the Adaptive Locator (Figure 6.1 on page 94) and the non-adaptive system (Figure 6.6) include the following:

- **Adaptive Locator** infers social connections between the user and other people. This in turn allows the system to denote the closeness of relationship with different shades of a colour on the map and to list only the people that the user is predicted to be acquainted with. By contrast, the non-adaptive system displays all the people found on that floor and denotes them with the same colour of circles on the map.

- **Adaptive Locator** may display personally meaningful location descriptions, such as *his office* or *Bob’s office*. In the case when a personalised label is displayed, the actual room or desk number is hidden from the user. In this case, the non-adaptive system would only display the room or desk number of a location, such as *Room 123* or *Desk 3W32*.

- **Adaptive Locator** allows the user to see explanations of the personalisation. This
may include showing evidence of location resolution, office ownership, social connections, and places that the user has recently been to.

6.2.3 Data Collection

The responses to the familiarisation tasks were collected, as one task was designed to determine how each participant would refer to a place which they were familiar with. This response did not affect the way that Adaptive Locator delivered the personalised information. For the system tasks, we collected the time each participant took to complete each task and the answers for the tasks. We used observation to assess whether participants completed the set tasks successfully. In addition, before they scrutinised the adaptive system’s explanation about personalised location labels, they were asked to think about what the system would need for such reasoning. We also kept a record of the participants’ activities (e.g., timestamp and positions of mouse clicks) with the map interface, explanations, and the link to show and hide less relevant people. For the system tasks and the questionnaire, the user input data was saved every 30 seconds to minimise data loss in the event of system or machine crash.

As for the non-adaptive system, the participants were asked to indicate, who of the 20 people shown by the system, they would, or would not, be interested in knowing locations of, and why. For the questionnaire, we aimed to collect the participants’ opinions on three aspects of the adaptive system: determining relevant people, displaying personalised location labels, and generating explanations about the personalisation (see Table 6.2 on page 102 for the actual questions). In particular we wanted to assess how satisfied they were with the personalisation, whether they would prefer a system that presents adaptive information, how understandable and useful the explanations were, and their perception about system scrutability.

6.3 Data Analysis and Results

6.3.1 Task Completion

Table 6.1 shows the mean times with standard deviations for participants to complete the two tasks for each system: (Task 1) locating an individual known to them and (Task 2) locating people in Level 4 West of the building. We used observation to assess whether participants completed the set tasks successfully. Time data needs to be interpreted cautiously as participants were not asked to work quickly and they tended to talk aloud as they used the system. There were no statistically significant differences in the time to
Data Analysis and Results

Table 6.1: Time of completion of the two tasks for the two systems. Data are in means ± standard deviations.

<table>
<thead>
<tr>
<th></th>
<th>Task 1 (seconds)</th>
<th>Task 2 (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-adaptive system</td>
<td>30 ± 12</td>
<td>24 ± 6</td>
</tr>
<tr>
<td>Adaptive system</td>
<td>44 ± 18</td>
<td>34 ± 19</td>
</tr>
</tbody>
</table>

All participants successfully completed all tasks, with both adaptive and non-adaptive systems.

6.3.2 Personalised Selection of People

We assessed the accuracy of the personalised selection of people by comparing (1) participants’ indication of whose location they wanted to know about, and (2) the personalised selection made by the system. Participants also rated their overall perceptions of the number of system mistakes on a seven-point Likert scale. Figure 6.7 shows the percentage of the actual (lighter line) and user perceived (darker line) system mistakes in displaying relevant people (see Q1 in Table 6.2), which average 24% and 25% respectively. Most chose the rating closest to the percentage of actual mistakes. D and F under-rated by one point on the seven-point scale, while A over-rated by one point. Participant A commented, “my need to find them [i.e. the people hidden by the system]
Table 6.2: Questions and responses of the post-study questionnaire

<table>
<thead>
<tr>
<th>User</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Avg</td>
<td>2.5</td>
<td>6.4</td>
<td>6.8</td>
<td>6.6</td>
<td>5.6</td>
<td>6.5</td>
<td>6.4</td>
<td>6.6</td>
</tr>
</tbody>
</table>

is less (often) than those that did show up”, hence the better rating than the actual system performance. Participants indicated that they preferred—with an average of 6.4 out of 7.0 rating—the personalisation (see Q2 in Table 6.2).

6.3.3 Personalised Location Label

In order to provide personalised labels for users, we analysed how users referred to a desk of a person they knew. Two tasks provided evidence.

1. The participants were asked whether they knew a desk location presented as a desk
number (e.g. Desk 3W32) and in the form of his desk respectively. All participants knew that location when given the personalised label, with the exception of one participant, who had not been to the place. Only one participant (let us call them P1 for later reference) recognised the desk location by a desk number, because P1 sat next to the desk. Two other participants knew the approximate location with only the desk number: one also sat next to the desk, and the other was a very close friend of P1.

2. Participants were asked how they would refer to the office of the co-leader of the research group. They unanimously put down “Judy’s office”. This confirms that people tend to associate locations with a more meaningful object or event than merely a location number (Ashbrook and Starner 2003; Hightower 2003).

Summarising Table 6.2, participants strongly agreed (see Q3 in the table) that personalised labels (e.g. Bob’s office) are more useful than the actual room number (e.g. Room 300). They all preferred to have personalised location labels in the system (Q4), unless they do not know the place (Q5). This result is notable in view of the fact that all participants were very familiar with the area and certainly know their own desk or room identifiers.

### 6.3.4 System Explanation

Regarding scrutability, participants indicated that the explanations were understandable (Q6) and that they explained what they wanted to know (Q7), except for the cases of hidden people, in the case of three participants. For that, the system’s explanation would be, for example, “You do not appear to know Alice”. This explanation was judged adequate unless the system had made a mistake. In that case, participants indicated they wanted more details of the reasoning process.

A strong rating of 6.6 was given to the importance of system explanation (Q8). The participant who gave the lowest rating of 5 explained that in some cases they would not care how the system achieved the personalisation.

### 6.4 Discussion

Even though the system calculates a confidence level of the social connection between the current user and each person found in the building, the system uses a minimal threshold: people would only be hidden if there is no evidence of any social connection with the user. One participant commented that “I knew [PERSON NAME], but as he
is an acquaintance, his location is not very relevant to me”, which implied the need for a higher or even an adaptive threshold for hiding people on the display.

One concern raised by five out of eight participants was the ability to control the information presented. This includes the ability to select who to show and hide, add missing evidence, and modify incorrect system beliefs. It was also observed that different degrees of control were needed. Two participants explicitly stated that they wanted the ability to manually select and configure which people to show and hide. One of them commented, “I would probably prefer to click a ‘don’t show me this person again’... and initially show everyone.”

The ability to know and scrutinise hidden information (e.g. people who were not displayed) was valuable, as stated by four participants. One commented, “I do like the ability to ask the system to show me the people who have been hidden, as it allows me to see...how full the labs are”. Another participant commented “[I prefer the adaptive system], as long as it is hiding info from me”.

Responding to Q4 in Table 6.2, two participants were neutral about whether a room number or a personalised label should be presented if they did not know the place. They argued that a personalised label could be more useful when trying to provide directions, or when they do not need to physically locate the person. One commented, “[Regardless of whether I know where Alice’s office is or not,] I would only want to know the office number if I need to physically find her.” Five participants indicated that they would like both the personalised location label and the room number to be displayed, e.g. Bob’s office (310).

Because of the nature of the goals of this work, we recruited a group of participants who worked in the same physical area. This may affect the rating shown in Table 6.2 on page 102, and, therefore, this limitation needs to be taken into account when generalising the results.

6.5 Summary

We have presented a user study to evaluate PECO that back-ends an adaptive system. In particular we assessed the following properties of the system:

- the accuracy and value of the selection of people relevant to the user;
- the value of personalised labels;
- the understandability of the explanations for the personalisation.
The study presented encouraging results:

- The adaptive system was able to achieve 76% and 75% on average for, respectively, actual and users’ perceived accuracy in inferring relevant people to display, even with limited evidence sources and a simple algorithm for reasoning about the social connections.

- Participants preferred a location system with this adaptation feature.

- Participants strongly preferred the personalised labels over location numbers and tended to associate a location with something personally meaningful, such as Judy’s office.

- The adaptive system was able to provide understandable and useful explanations of personalisation using a personal ontology.

- Participants considered it important to have explanations for system adaptation.

These conclusions need to be interpreted in light of the particular population who participated in the study. Even so, these results reflect the potential for a personal ontology to empower an adaptive system with personalisation and scrutability. Moreover, the results indicate that the PERSONAF framework, as we have implemented it, supports ontological reasoning for scrutable personalisation of information about a pervasive computing environment.
Chapter 7
Conclusions

This thesis aims to explore ways to create and exploit a personal ontology for indoor pervasive computing environments to tackle three critical issues: conflict resolution for sensor fusion, personalised information delivery, and explanation of personalisation in adaptive systems. We now describe how each chapter makes contributions discussed in Chapter 1 to achieve this overall goal.

Chapter 3 describes the Personalised Pervasive Scrutable Ontological Framework (PERSONAF), which consists of two main components: a central ontological knowledge base—the Personalised Context Ontology (PECO)—and a reasoning component—the Ontology- and Evidence-based Context Reasoner (ONCOR). It adopts the accretion and resolution approach for the accretion ontology layer of PECO and the central mechanism of ONCOR, which consists of a number of basic and complex context resolvers, with special support for ontological location resolution. The contributions of this chapter are:

- The conceptual model for a framework, PERSONAF, to support modelling of entities and ontological reasoning about location information in pervasive computing environments;

- Presenting three substantially different ontological approaches to reason about locations, including detection and resolution of conflicting sensor evidence.

Chapter 4 presents our implementation of the PERSONAF framework in an indoor pervasive computing environment. This involved choosing an appropriate top-level ontology and representation to build a middle ontology to model the interior space of a building; choosing an authoritative domain-specific document to form a base application ontology, by extending the middle ontology; selecting a range of sources to mine, in order to populate the application ontology with relevant propositions, which then
forms the novel personalisation layer of PECO. We also reported a preliminary evaluation of PECO, based on validations of its syntax and hierarchy and a comparison-based evaluation. Key contributions are:

- Constructing the personal ontology based upon a hybrid approach and a range of heterogeneous relevant sources;
- Designing a middle ontology to model the inside of a building by extracting concepts and structures from a top-level ontology, OpenCyc (Lenat 1995), where this ontology facilitates reusability and interoperability, and it forms the basis for population of light-weight ontologies;
- Demonstrating ways to learn and populate a context ontology with a range of domain-specific sources;
- Creating reusable tools to build an application ontology and an accretion ontology within the PERSONAF framework.

Chapter 5 evaluates the upper two layers of PECO—the middle ontology and the application ontology—in terms of its ability to detect and resolve conflicting location evidence across different levels of location granularity. The results indicated the capability of an ontology in location conflict resolution. This suggests a great potential for an ontology to play a pivotal role in assisting systematic conflict resolution in pervasive computing applications. The key contributions are:

- Demonstrating the capability of an ontology to detect and resolve conflicting information in pervasive computing;
- Conduct of a careful experimental evaluation which is based on reliable records of people’s movements within a building.

Chapter 6 describes the Adaptive Locator system back-ended by the PERSONAF framework and its qualitative evaluation. The user study evaluates the adaptive system in terms of its ability to deliver correct and useful personalisation, as well as to generate understandable explanations of the personalisation. The results indicated that the system was able to deliver useful personalisation with accuracy 76% on average, and that it was able to provide explanations that study participants rated as understandable and useful explanations both for the personalisation using the implemented personal ontology and its underlying reasoning. The key contributions are:
• Showing that a personalised context ontology could be used in personalised information delivery, in terms of adaptively selecting relevant people and personalising location descriptions;

• Illustrating the capability of our personalised context ontology and underlying framework in generating comprehensible explanations of the personalisation.

Future Work

One critical aspect of pervasive computing that this thesis has not specifically addressed is privacy control (Langheinrich 2001). There is a small amount of work on use of ontologies in privacy management. For example, Hong and Landay (2004) proposes a privacy control technique by abstracting away the finest granularity of location information. Wishart, Henricksen, and Indulska (2007) extends the technique to obfuscate other contextual information.

Another possible approach to privacy control is to make use of a personal ontology to present adaptive location labels, which may have the effect of privacy by obscurity (Eymann and Morito 2004). So, for example, Alice might be willing to release her current location as my favourite coffee shop. This immediately affords privacy about her location in relation to people who do not know enough about her to map this to her actual location. To account for the rapid-changing pervasive computing environment, this approach must be able to easily adapt to changes, such as when Alice changes her favourite coffee shop.

For the Adaptive Locator, there are several aspects to extend the personalised information delivery, some of which were suggested by the study participants.

• Select relevant people based on some kind of social groups, such as people sitting nearby me, project collaborators, and my students.

• Use a more flexible threshold to select relevant people to display. For example, instead of showing everyone the user is inferred to know, it may help to hide some people who they know less well.

• Base the selection of people on activity, as there has been work on activity detection and recognition (Kaptelinin 2003; Dragunov et al. 2005; Bardram et al. 2006). So, for example, when Bob is busy writing a paper to meet a tight deadline, he might only want to see his co-authors’ whereabouts.
• Adaptively adjust the number of people to be shown based on the clutterness of the display.

Such a system might also be extended to enhance social interaction, for example, incorporating services like location-based online chat and delivery of short messages via Bluetooth.

In regards to location resolution, future work may incorporate the reasoning between time and distance. So, for example, by accounting for one’s walking distance measure, it may help eliminate noisy location data.

Our work reported two user studies. Both are limited in terms of the small number of participants and their scope. At this stage of research, these qualitative studies provide valuable insights. However, it will be important, in the future to learn from more extensive studies and deployments. It would also be interesting to see how this framework can be generalised in another domain, such as a heritage building or a museum.

Summary

This thesis presents a proposal, an implementation, and two user evaluations of a conceptual framework, PERSONAF, to model and reason about the pervasive computing environment, where personalisation is dependent on rapidly changing contexts. Central to the framework is a layered model of a personalised context ontology, or PECO, that consists of a middle ontology, an application ontology, and an accretion ontology, which provides the novel personalisation layer. We described how each layer of PECO is implemented, and how it is interpreted with ONCOR, the reasoning component within PERSONAF. Then we evaluated the framework with two user studies:

• One evaluated the application ontology for a building that extends MIBO to reason about conflicting location information in a multi-sensor environment.

• The other one evaluated PECO for its capability to deliver personalised information and understandable explanation of personalisation.

Both empirical studies yielded promising results, which indicate the important role that an ontology will likely to play in pervasive computing in modelling and reasoning about context.
Bibliography


A. Agostini, C. Bettini, and D. Riboni. Experience report: Ontological reasoning for context-aware internet services. In Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06), pages 8–12, Mar. 2006. 1


J. Hartmann, Y. Sure, A. Giboin, D. Maynard, M. del Carmen Suárez-Figueroa, and R. Cuel. Methods for ontology evaluation. KWeb Deliverable D1.2.3, University of Karlsruhe, Dec 2004. 15


J. Hightower. From position to place. In *Proc. of The Workshop on Location-Aware Computing*, 2003. 4, 28, 103


J. Kay, A. Lum, W. Niu, and L. Li. TARO: Teaching with an automatically retrieved ontology. Technical Report 596, University of Sydney, September 2006. 50


## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/R</td>
<td>Accretion and Resolution</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>BDI</td>
<td>Believes, Desire, Intention</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CCS</td>
<td>Closest Common Subsumer</td>
</tr>
<tr>
<td>COBRA</td>
<td>Context Broker Architecture</td>
</tr>
<tr>
<td>COBRA-ONT</td>
<td>Context Broker Architecture Ontology</td>
</tr>
<tr>
<td>CONON</td>
<td>Context Ontology</td>
</tr>
<tr>
<td>DAML</td>
<td>DARPA Agent Markup Language</td>
</tr>
<tr>
<td>DL</td>
<td>Description Language</td>
</tr>
<tr>
<td>FIPA</td>
<td>Foundation for Intelligent Physical Agents</td>
</tr>
<tr>
<td>FOAF</td>
<td>Friend of a Friend</td>
</tr>
<tr>
<td>FVP</td>
<td>Frequently Visited Places</td>
</tr>
<tr>
<td>GH</td>
<td>Granularity Harmoniser</td>
</tr>
<tr>
<td>GUMO</td>
<td>General User Model Ontology</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>LocLog</td>
<td>Location Log</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MIBO</td>
<td>Middle Building Ontology</td>
</tr>
</tbody>
</table>
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MILO</td>
<td>Mid-Level Ontology</td>
</tr>
<tr>
<td>ONCOR</td>
<td>Ontology- and Evidence-based Context Reasoner</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>PECO</td>
<td>Personalised Context Ontology</td>
</tr>
<tr>
<td>RCC</td>
<td>Regional Connection Calculus</td>
</tr>
<tr>
<td>SIT</td>
<td>School of Information Technologies</td>
</tr>
<tr>
<td>SKOS</td>
<td>Simple Knowledge Organisation System</td>
</tr>
<tr>
<td>SOUPA</td>
<td>Standard Ontology for Ubiquitous and Pervasive Applications</td>
</tr>
<tr>
<td>SUMO</td>
<td>Suggested Upper Merged Ontology</td>
</tr>
<tr>
<td>SVG</td>
<td>Scalable Vector Graphics</td>
</tr>
<tr>
<td>U2M</td>
<td>Ubiquitous User Modelling</td>
</tr>
<tr>
<td>ULCO</td>
<td>Upper-level Context Ontology</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Appendix A

MIBO and SIT Ontology in OWL

A.1 MIBO

<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:p1="http://www.cyc.com/2004/06/04/cyc#"
    xmlns:p2="http://www/owl-ontologies.com/assert.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    <owl:Ontology rdf:about=""/>
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">$Id: v0.2 $</owl:versionInfo>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      This ontology is derived from the OpenCyc v0.7.8b made available on the opencyc.org website. The following is the copyright information for the original ontology.
      Copyright Information OpenCyc Knowledge Base Copyright 2001-2004 Cycorp, Inc., Austin, TX, USA. All rights reserved. OpenCyc Knowledge Server Copyright 2001-2004 Cycorp, Inc., Austin, TX, USA. All rights reserved. Other copyrights may be found in various files. The OpenCyc Knowledge Base The OpenCyc Knowledge Base consists of code, written in the declarative language CycL, that represents or supports the representation of facts and rules pertaining to consensus reality. OpenCyc is licensed using the GNU Lesser General Public License, whose text can also be found on this volume. The OpenCyc CycL code base is the "library" referred to in the LGPL license. The terms of this license equally apply to renamings and other logically equivalent reformulations of the Knowledge Base (or portions thereof) in any natural or formal language. See http://www.opencyc.org for more information.</rdfs:comment>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:RDF>
A.1. MIBO

> Middle Building Ontology <rdfs:label>
</owl:Ontology>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#Building"/>
<owl:Class rdf:id="SpaceInAFixedStructure">
  <rdfs:subClassOf>
    <owl:Class rdf:id="Place"/>
  </rdfs:subClassOf>
  <owl:equivalentClass>
    <owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#SpaceInAFixedHOC"/>
  </owl:equivalentClass>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    #$SpaceInAFixedHOC is a specialization of #$SpaceInAHOC and #$Place. Each instance of
    #$SpaceInAFixedHOC is a space within some instance of a
    fixed (as opposed to moveable) #$HumanOccupationConstruct
    (q.v.). Positive examples include rooms within #$Buildings,
    while negative examples include spaces within recreational vehicles.
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#Place-NonAgent"/>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#MultiRoomUnit"/>
<owl:Class rdf:id="Elevator">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:id="Building"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:TransitiveProperty rdf:id="isPartOf"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:equivalentClass>
    <owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#ElevatorCar"/>
  </owl:equivalentClass>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    #$ElevatorCar is a specialization of #$Elevator.
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#Stairwell"/>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#OfficeSpace-Personal"/>
<owl:Class rdf:about="http://www.cyc.com/2004/06/04/cyc#RoomInAConstruction"/>
<owl:Class rdf:id="MultiRoomUnit">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:id="LevelInABuilding"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
A suite, bunch of rooms, (or even just one single room) which are thought of as one unit. #$MultiRoomUnits can have rooms.

A specialization of both #$FixedStructure and #$HumanShelterConstruction. Each instance of #$Building is
A.1. MIRO

A (usually large) fixed structure with walls and a roof, and with some inside area or areas designed to be occupied by humans (but not necessarily as a residence). Examples include the Empire State Building, Hearst Castle, an aircraft hangar at O'Hare, a lighthouse in the Mediterranean sea, the Sydney Opera House, and the Washington Monument in Washington DC.
APPENDIX A. MIBO AND SIT ONTOLOGY IN OWL

A.2 SIT Ontology

<?xml version="1.0"?>
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about=""/>
  <owl:Ontology rdf:about="http://www.w3.org/2000/01/rdf-schema#"/>
  <owl:Ontology rdf:about="Spatial ontology for the School of IT, University of Sydney"/>
</owl:Ontology>
A.2. SIT ONTOLOGY

<mibo:Building rdf:ID="sit_building">
  <mibo:hasPart>
    <mibo:LevelInABuilding rdf:ID="sit_level1">
      <mibo:hasPart>
        <mibo:MultiRoomUnit rdf:ID="sit_level1west">
          <mibo:hasPart>
            <mibo:SingleRoom rdf:ID="sit_111">
              <mibo:isPartOf rdf:resource="#sit_level1west"/>
            </mibo:SingleRoom>
          </mibo:hasPart>
          <mibo:hasPart>
            <mibo:SingleRoom rdf:ID="sit_119">
              <mibo:isPartOf rdf:resource="#sit_level1west"/>
            </mibo:SingleRoom>
          </mibo:hasPart>
          ...
          ...
          <mibo:hasPart>
            <mibo:SingleRoom rdf:ID="sit_114">
              <mibo:hasPart>
                <mibo:PersonalOfficeSpace rdf:ID="sit_114d01">
                  <mibo:isPartOf rdf:resource="#sit_114"/>
                </mibo:PersonalOfficeSpace>
              </mibo:hasPart>
              <mibo:hasPart>
                <mibo:PersonalOfficeSpace rdf:ID="sit_114d02">
                  <mibo:isPartOf rdf:resource="#sit_114"/>
                </mibo:PersonalOfficeSpace>
              </mibo:hasPart>
              ...
              ...
              <mibo:isPartOf rdf:resource="#sit_level1west"/>
            </mibo:SingleRoom>
          </mibo:hasPart>
          ...
          ...
          <mibo:isPartOf rdf:resource="#sit_level1"/>
        </mibo:MultiRoomUnit>
      </mibo:hasPart>
    </mibo:LevelInABuilding>
  </mibo:hasPart>
</mibo:Building>
<mibo:hasPart>
  <mibo:MultiRoomUnit rdf:ID="sit_level1middle">
    <mibo:hasPart>
      <mibo:Elevator rdf:ID="sit_l100">
        <mibo:isPartOf rdf:resource="#sit_level1middle" />
      </mibo:Elevator>
    </mibo:hasPart>
    ... ...
    <mibo:isPartOf rdf:resource="#sit_level1" />
  </mibo:MultiRoomUnit>
</mibo:hasPart>
<mibo:isPartOf rdf:resource="#sit_building" />
<mibo:LevelInABuilding>
</mibo:hasPart>
</mibo:LevelInABuilding>
<mibo:hasPart>
  <mibo:MultiRoomUnit rdf:ID="sit_level2middle">
    <mibo:isPartOf rdf:resource="#sit_level2" />
  </mibo:MultiRoomUnit>
</mibo:hasPart>
<mibo:LevelInABuilding rdf:ID="sit_level2">
  <mibo:hasPart>
    <mibo:MultiRoomUnit rdf:ID="sit_level2west">
      ... ...
    </mibo:MultiRoomUnit>
  </mibo:hasPart>
  <mibo:hasPart>
    <mibo:MultiRoomUnit rdf:ID="sit_level2east">
      ... ...
    </mibo:MultiRoomUnit>
  </mibo:hasPart>
  <mibo:LevelInABuilding>
</mibo:hasPart>
</mibo:LevelInABuilding>
<mibo:hasPart>
  <mibo:MultiRoomUnit rdf:ID="sit_level2middle">
    <mibo:isPartOf rdf:resource="#sit_level2" />
  </mibo:MultiRoomUnit>
</mibo:hasPart>
<mibo:LevelInABuilding rdf:ID="sit_level3">
  <mibo:hasPart>
    <mibo:MultiRoomUnit rdf:ID="sit_level3west">
      <mibo:hasPart>
        <mibo:SingleRoom rdf:ID="sit_329f">
          <mibo:isPartOf rdf:resource="#sit_level3west" />
        </mibo:SingleRoom>
      </mibo:hasPart>
      ... ...
      <mibo:PersonalOfficeSpace rdf:ID="sit_3w11">
        <mibo:isPartOf rdf:resource="#sit_level3west" />
      </mibo:PersonalOfficeSpace>
    </mibo:MultiRoomUnit>
  </mibo:hasPart>
</mibo:LevelInABuilding>
A.2. SIT ONTOLOGY

<!--Generated by svg2mibo.py-->
Appendix B

PersonisAM Default Resolvers

This appendix describes the default resolvers in PersonisAM that have been used in ONCOR (see Section 3.4 on page 38).

**MostRecentEvidenceValue**  This resolver simply returns the value of the latest evidence. It is the simplest and most computation-efficient resolver function. It is used in resolving values that rarely alter, such as a person’s name and a device’s Media Access Control (MAC) address.

**MostRecentEvidence**  This resolver returns the most recent piece of evidence, which may include information like a value, an evidence source, and a timestamp of the evidence.

**CountedEvidence**  This resolver returns a list of evidence according to the given number, or ‘all’.

**RecentEvidenceTimed**  This resolver returns a list of all evidence that is younger than a given time in seconds.

**RecentEvidenceTimedWithinLatest**  This resolver returns a list of the most recent evidence that is within a given time in seconds from the latest evidence time.

**RecentUniqueValues**  This resolver returns a list of unique evidence values that are younger than a given time seconds.
Appendix C

Sensor Messages

The following table is a summary of the diverse range of devices and sensors used for the eight participants over the period of the LocLog experiment. Columns represent the sensors that contributed to the evidence sources: login sensor (LG), system activity sensor (SYS), Bluetooth sensor (B). Rows indicate the participants’ devices used during this experiments. So, for instance, participant A’s phone was detected 66 times by Bluetooth sensor one (B1), and the activity sensor (SYS) on they desktop computer had transmitted 2537 Present or Lost messages over the experiment period. There were also two records from a login sensor for that participant indicating that they had logged into a machine in the computer lab in Level 1.
## Appendix C. Sensor Messages

| A | phone | 66 | 388 | 11 | 111 | 37 | 139 | 61 | 3290 | 40 | 307 | 4 | 118 | 71 | 649 | 29 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | tablet PC | 56 | 329 | 11 | 69 | 18 | 81 | 64 | 3948 | 32 | 190 | 4 | 318 | 47 | 298 | 38 |
| C | desktop | 2537 |
| | laptop | 2629 |
| | other | 2 |
| D | phone | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
| | tablet | 0 | 47 | 0 | 13 | 0 | 0 | 28 | 1352 | 0 | 35 | 0 | 326 | 21 | 27 | 0 |
| | desktop | 1071 |
| | laptop | 380 |
| E | tablet | 33 | 84 | 0 | 0 | 0 | 6 | 0 | 8 | 0 | 37 | 0 | 0 | 34 | 2 | 874 |
| | desktop | 971 |
| F | phone | 0 | 105 | 44 | 52 | 2 | 28 | 28 | 846 | 4 | 123 | 4 | 120 | 201 | 2 |
| | tablet | 0 | 0 | 0 | 0 | 0 | 0 | 577 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | laptop | 0 | 106 | 20 | 33 | 88 | 12 |
| | other | 0 | 0 | 0 | 0 | 0 | 0 | 28 | 123 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | tablet | 0 | 39 | 0 | 0 | 0 | 27 | 0 | 860 | 0 | 8 | 0 | 0 | 2 | 0 |
| | desktop | 887 |
| H | phone | 0 | 2 | 0 | 40 | 480 | 2 | 0 | 7 | 4 | 118 | 4 | 118 | 71 | 649 | 29 |
| | tablet | 0 | 16 | 0 | 4 | 808 | 24 | 1 | 69 | 18 | 81 | 64 | 3948 | 32 | 190 | 4 |
| | desktop | 718 |
| | laptop | 722 |
| | other | 2 |
| Total | 49 | 13252 | 234 | 1233 | 88 | 328 | 2067 | 363 | 360 | 12297 | 102 | 873 | 8 | 952 | 333 | 1368 | 2944
Appendix D

Questions and Tasks for PECO User Study

Start-up Questions

1. Full name

2. What is your room/desk number (e.g. 310, 3w14)?
Interface Familiarisation Tasks

This section aims to familiarise you with the map interface you will see in the rest of the study. There will be short tasks in this section. The estimated time of completion is 2 to 5 minutes.

1. The following is the Level 3 floor map, the level where your office is located, of the School of IT building. If the red dot on the map presents a person, how would you refer to the place where the person is located?

Above is an interactive map. For the following tasks, please use your mouse to click on the appropriate places on the map.

2. Suppose the red dot is Judy, where is she?

3. Where is your office?

4. Where is Room 350?
Tasks for the Non-adaptive System

At this page, you will be asked to do tasks with the system. The estimated time of completion is 5 minutes.

Click here to open up the system in a new window. After the system has started up, please continue to the next task.

Please answer the following tasks according to the system.

1. Where is X [someone whom the system inferred as a relevant person to the user]?

2. Would you have known X’s location only from the system’s textual description (i.e. [answer of question 2])? [yes/no/maybe]

3. Who is at Level 4 East?

4. Please go to each level and take a look at the people displayed. Who or what groups of people would you have not wanted to see, i.e. you are not interested in knowing their locations? Please give reasons. (e.g. Bob - I would only want to see him when a group meeting is coming up)
Tasks for the Personalised System

At this page, you will be asked to do tasks on a more personalised version of the system. The estimated time of completion is 5 to 10 minutes.

Click here to open up the system in a new window. After the system has started up, please continue to the next task.

Please answer the following tasks according to the system.

1. Where is X [someone whom the system inferred as a revelant person to the user]? 

2. Would you have known X’s location only from the system’s textual description (i.e. [answer of question 2])? [yes/no/maybe]

3. Who is at Level 4 East?

4. What kind of information do you think the system needs to gather in order to display ’X’s desk’ instead of for X’s location?

5. Find out what evidence the system gathered in order to display ’X’s desk’ for X’s location.

6. Find out why the system displays ’Judy Kay’ (on Level 3) to you.

7. Find out why the system hides ’Pierce Chen’ (on Level 4) from you.
Post-study Questionnaire

There are questions regarding the two systems you just viewed. Feel free to go back to each system when you answer these questions.

Each answer is based on a 7-point scale, where 1 means complete disagreement and 7 means complete agreement. You are encouraged to comment on the reasons for the point you give to each question. The estimated time of completion is 10 to 15 minutes.

Personalisation

1. The personalised system made a lot of mistakes in terms of displaying relevant people to you and hiding irrelevant ones from you.

<table>
<thead>
<tr>
<th>disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Enter your comments here

2. If the personalised system can correctly display the relevant people and hide the irrelevant ones, you would prefer to have that feature than not.

<table>
<thead>
<tr>
<th>disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Enter your comments here

3. Personalised labels (e.g. 'Bob’s office’) are more helpful than room/desk numbers (e.g. 'Desk 3W32’).

<table>
<thead>
<tr>
<th>disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Enter your comments here
4. If the personalised system can correctly display personalised location labels for the places you know about, you would prefer to have that feature than not.

System’s Explanations

1. The explanations generated by the system were understandable.

2. The explanations provided by the personalised system told you what you wanted to know. (If not, please comment on what you wanted to know instead.)
3. It is important to have explanations for the personalisation.

<table>
<thead>
<tr>
<th>disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>❌</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter your comments here
Demographic Questions

1. Age:

2. Gender:

3. Role in uni: [staff member/visiting scholar/postgraduate student/undergraduate student/other]

4. Are you a native English speaker?

5. How frequent do you use the Locator system? [multiple times a day/1-5 times a week/1-5 times a month/never used it]

6. If you have never used Locator before, have you heard about it or seen the interface? [heard about it/seen the interface/both/neither]

7. How long have you been working in this building? [since it was opened/more than 6 months/1–6 months/2-4 weeks/less than 2 weeks]