Hasselt University
and transnational University of Limburg

The Variability of Location Context in Pervasive Environments: Modelling, Representation and Visualisation

Dissertation submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science
at Hasselt University
to be defended by

Petr Aksenov

Advisor: prof. dr. Karin Coninx
Co-advisor: prof. dr. Kris Luyten

Nov 2007 – Jun 2012

Acknowledgements

So here it comes – the moment to look back at what I have been through in the past few years: from the very first day, when I opened the doors of a place that would become my second – and sometimes first, literally! – home, to the moment of typing this text. Something has been achieved and realised, somewhat more has been discussed and tried out, but a lot more has been learned. I have been surrounded with people who have been out there to support me professionally, culturally, socially, personally, and anyhow else that I was walking with side by side while taking this journey; and who have helped me – in either way – make this PhD what it has become.

Without exaggeration, I am happy to say that I am most grateful to my advisor, prof. Karin Coninx, and my co-advisor, prof. Kris Luyten; and in the first place for letting me take this opportunity to pursue a degree. In the years I spent at EDM under Karin's and Kris' supervision, I gradually learned the ways things should be approached in academia. The feedback I received from them on the initial ideas I expressed, the motivation and encouragement I was given to delve into those worth exploring, and the numerous discussions we had regarding the flow of my activities made my progress evolve in a smooth and proper way. Besides, their feedback on the style of my academic writing cannot be overestimated, for it was priceless indeed.

I greatly appreciate that dr. Jan Van den Bergh, prof. Wim Lamotte, prof. Peter Quax, prof. Filip De Turck, and prof. Khalid Al-Begain accepted the invitation to be in the jury; and that prof. Marc Gyssens agreed to serve as the jury chair.

I would like to express my sincere gratitude to my colleagues – dr. Mieke Haesen, dr. Jan Meskens, Davy Vanacken, dr. Geert Vanderhulst, Jo Vermeulen, and Anastasiia Beznosyk, dr. Nasim Mahmud, dr. Johanna Octavia, Dawid Ostrowski, Miquel Rujula, Deepak Sahni, Jan Schneider, Sean Tan, Joël Vogt – with whom I had a pleasure to share the working environment and communicate and discuss on a great deal of work- and science-related and -unrelated topics as well as on multiple aspects of life in Belgium and beyond.

I would like to send my special thanks to Tom De Weyer and Peter Quax at EDM and to Andres Henckens and his colleagues at CID, whose help and patience in arranging the experimental and technical facilities that I needed during my activities went beyond measure.

I want to thank Karel Robert for his analysis and suggestions on how the details behind my research ideas should best appear to the end-users; and I am grateful to each and every participant of the user studies that I performed while validating these things.

During my time at EDM, I was a member of the CoLaSUE project, a research initiative funded by the Research Foundation Flanders (F.W.O. Vlaanderen, project number G.0439.08N). In this regard, I want to mention our project collaborators from the DistriNet research group at KULeuven – Ansar-Ul-Haque Yasar, Davy Preuveneers, and Yolande Berbers – with whom we had many productive sessions and moments of wisdom. Without their valuable feedback and experienced advice shared during our fruitful discussions, an important part of this dissertation would not have appeared.

Apart from science, I was happy to spend the invaluable moments during my out-of-office hours in the warm company of Aleksandr Kuznetsov, Pavel Poliakov, Gianpaolo Papari, Michael Vorndran, José Proença and Ksenia Rogova, Georgy Pashkov and Olga Ivanchikova, Dorin Cerbu, Sergey Deutsch, Yana Chernichenko, Stanislav Popkov, Kaushik Rajaram, and many others.

I send a separate thank to dr. Pavel Galibarov, my greatest friend and my big brother in the life of a doctoral student. Our hot disputes about (our) science(s), discussions about world news, opinions about the life in Europe and plans for the future, and about a whole bunch of other topics – all kept me staying connected. And his words of cheer that I heard in the difficult moments of writing and evaluating my achievements gave me power and strength to continue.

And of course, I will always remain thankful to my dearest family who always supported me anyhow they could. A huge, though invisible, part of this dissertation is their achievement, too.

Abstract

Context and context-awareness are crucial concepts in pervasive computing. Among the many and diverse types of contextual information, information about location is generally seen as one of the most important. There is a great deal of situations the users happen to be in throughout a pervasive environment. This fact imposes restrictions on the use of the available resources and the surroundings as well as requires taking into account the users' personal preferences and requested customisations. As a result, location context undergoes permanent and often considerable changes over time and space: e.g., loses existing location data, faces changes in quality of these data, or receives unknown data. In other words, location context in a pervasive environment is exposed to high variability, which puts limitations on the applicability of a pervasive application that is expected to use this context.

This dissertation explores how some aspects of the variability of location context in a pervasive environment could be approached in a way suitable to such environment and utilised therein. We introduce a user-centric model of spatial arrangements around as well as between resources in the environment with a special focus on the model's ease of reference and manipulation at all stages of an application's development and use, and with support of uncertain and incomplete knowledge. We show how the proposed model can be integrated into a framework for dealing with pervasive environments, thereby making the framework capable of managing location-based tasks.

Further on, we present an approach to coping with a multitude of location determination technologies by introducing a unified view on the existing diversity of location sensing data and formats in terms of their representation and processing within the application. We provide the approach with a number of software tools and instruments that aim at easing the inclusion of an arbitrary location provider.

We then use the proposed unified approach as the underlying mechanism to facilitate our investigation of user needs in awareness of location context variability. Based on two real-life user studies throughout a medium-to-large-scale environment, we identify users' preferences regarding knowing about their location tracking conditions and come up with a number of design guidelines and implications on visualising location context variability, which can be taken into account when developing location-aware applications.

Finally, we demonstrate the benefits of combining location context with other types of context. By combining information about run-time location of members of a vehicular ad-hoc network, i.e. vehicles, with personal preferences of their drivers, we achieve a more efficient data dissemination scheme within that network and manage to deliver more relevant information as compared to the state-of-the-art algorithms.

Samenvatting

Context en context-awareness zijn cruciale begrippen in pervasive computing. Informatie over locatie wordt over het algemeen gezien als een van de belangrijkste vormen van contextuele informatie. In een pervasive omgeving is er meestal een hele waaier aan situaties waarin gebruikers zich bevinden. Dit brengt met zich mee dat er beperkingen zijn betreffende het gebruik van de beschikbare middelen in een omgeving. Bovendien vereist dit dat er rekening wordt gehouden met de gebruikers en hun persoonlijke voorkeuren. Contextuele informatie betreffende locatie ondergaat continue en vaak aanzienlijke veranderingen doorheen tijd en ruimte waarbij verlies van bestaande locatiegegevens, veranderingen in de kwaliteit van deze gegevens, of ontvangst van onbekende gegevens kan voorkomen. Met andere woorden, locatiegegevens in een pervasive omgeving worden blootgesteld aan een hoge variabiliteit, die beperkingen stelt aan de toepasbaarheid van pervasive applicaties die gebruik maken van dit soort contextuele informatie.

Dit proefschrift onderzoekt hoe een aantal aspecten van de variabiliteit van locatiecontext op een gepaste wijze kunnen worden benaderd en toegepast in pervasive omgevingen. We introduceren een gebruikersgericht model van ruimtelijke arrangementen rond en tussen bronnen in een omgeving. Hierbij ligt de focus op het gemak waarbij het model kan worden aangewend in alle stadia van de ontwikkeling en het gebruik van een toepassing, rekening houdend met onvolledige of onjuiste gegevens. We verduidelijken hoe dit voorgestelde model kan worden geïntegreerd in een raamwerk voor het beheer van pervasive omgevingen zodat dit raamwerk in staat is om locatie gebaseerde taken te beheren.

Verder presenteren we een aanpak voor het omgaan met een scala aan plaatsbepalingtechnologieën door een eengemaakt beeld te geven van de bestaande diversiteit aan data en formaten voor locatiebepaling en hun toepasbaarheid in applicaties. Wij ondersteunen deze aanpak aan de hand van een aantal software tools en instrumenten die het gebruik van willekeurige plaatsbepalingtechnologieën vergemakkelijken.

De voorgestelde eengemaakte aanpak wordt gebruikt als onderliggend mechanisme om ons onderzoek naar de gebruikersnoden betreffende het besef van locatie context variabiliteit te vergemakkelijken. Op basis van twee gebruikersstudies in een middelgrote tot grote omgeving, worden gebruikersvoorkeuren met betrekking tot kennis over locatie context variabiliteit geïdentificeerd. Vervolgens worden een aantal ontwerprichtlijnen en gevolgen voor het visualiseren van locatie context beschreven, die in aanmerking kunnen komen voor locatie gebaseerde toepassingen.

Tot slot tonen we de voordelen van het combineren van locatie context met andere vormen van context. Door het combineren van informatie over de huidige locatie van leden van een ad-hoc voertuignetwerk, met de persoonlijke voorkeuren van hun chauffeurs, komen we tot een meer efficiënte verspreiding van gegevens binnen dat netwerk en slagen we erin om relevantere informatie te voorzien in vergelijking met de state-of-the-art algoritmen.

Table of Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

1.1 Pervasive Computing

Succeeding the times of mainframe, mini- and personal computing, pervasive computing is a fourth, and today's, era of view on the relationships between humans and computers (Ye, Dobson, and Nixon, 2008). It is a post-desktop interaction paradigm based on Mark Weiser's vision where technologies are integrated into the environment and made invisible and indistinguishable in people's everyday life (Weiser, 1991).

Interaction in a pervasive computing environment should happen unobtrusively, with applications adapting their behaviour to the given situation. Such adaptation relies on an application's ability to extract and exploit relevant information in the environment; this information is referred to as context, and the ability to adapt is called context-awareness (Schilit and Theimer, 1994).

1.1.1. Context and context-awareness

According to Dey and Abowd (2000),

> "Context is any information that can be used to characterize 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."

> "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

Thus, context is a truly multifaceted and complex concept, and contribution of each type of context in the existence of a context-aware application is determined by this application's needs. Dey (Dey, 2009) talks about four types of context that are considered to be more important than the rest: location (where?), identity (who?), time (when?), and activity (what?). Using the information provided in these four context types, one can obtain various further information considered relevant to the situation in question. For instance, given a person's location, we can find out about presence of other people at and around that location and what they are doing.

Today, with the initial vision of pervasive computing approaching a reality, the understanding of the notion of context is expanding. Research efforts in this field shift currently from context aware problems to socially aware and adaptive computing, focusing on user activities and targeting large-scale systems interacting with communities of users (Lukowicz, Pentland, and Frescha, 2012). On the other hand, the success of location-aware applications and location-based services, which have spread into almost any aspect of today's life, still keeps the location context being the topic of a considerable amount of research efforts (Dey et al., 2010).

1.1.2. Location as context

Ye (2009) mentions three levels in the framework of a typical context-aware system: a sensor level that produces context, a context level that stores and manages the context obtained from the sensor level, and an application level that uses context. Speaking about the location context, the sensor level becomes responsible for the process of location determination and tracking. The context level applies – in accordance with the involved location model and the ways to manipulate the sensed context – the sensed location data to describe the state of the spatial world. And the application level uses the location model to determine appropriate actions. Each level has its own specifics and contains open questions to focus on from the perspective of a pervasive environment.

Sensor level

Although the importance of location context has caught a lot of attention, there is still no single technology for location determination that would meet all the requirements of the many scenarios where the knowledge about location and other spatial arrangements is required (Varshavsky and Patel, 2009). Such factors as the accuracy, installation costs, the environment infrastructure, or the required coverage area, to name just a few, have caused the creation of a diversity of technologies with which location can be determined. Usually, a suitable technology is chosen on the basis of a particular application's needs. As a result, in a heterogeneous pervasive environment, the availability and consistency of location information everywhere and at any time is not guaranteed. Therefore additional efforts are necessary to make the many location sensing techniques contribute to the availability and accessibility of location throughout the entire environment.

Context level

The location model underlying the context level is a representation of knowledge about objects' locations and spatial relationships that exist between these objects in the environment. Locations can be represented in a numerical, a symbolic, or another suitable form, and relationships such concepts as the distance between objects, a way to determine what is close/far or to determine that an object is situated to the left/right, to derive when a pair of objects happen to be in the same area or going in the same direction, etc. The location model can be extended with measures that would allow us to determine that one object is, for example, farther or moving faster than the other, and so on. Many representations exist; the suitability of each model is defined by the application needs, and the applicability of the model is linked to the available technologies for obtaining location information. This multifacetedness all together has an impact on the amount of information the model possesses, such as whether required relationships can be derived or the knowledge at hand is insufficient to make the exact required judgement. In the latter case we talk about uncertainty of location context. This uncertainty is an intrinsic part of the location context, which, given the heterogeneity of a pervasive environment, plays a yet more important role, and therefore must be handled with due care.

Application level

Dey and Abowd (2000) specify three categories of features that a context-aware application may support: 1) automatic *execution* of a service, 2) *presentation* of information and services to a user, and 3) *tagging* of context to information for later retrieval. Within location context, we understand the three categories as three types of situations in which location context can be exploited by a pervasive application. The execution is to adapt the application's behaviour to the available location conditions, the presentation is to make users aware of the available location conditions and changes, and the tagging is to attach location context to other context, i.e. to combine location and other types of context, so as to address a wider range of tasks.

1.2. Research challenges and contributions

Mark Weiser described pervasive computing[1] as "invisible, everywhere computing" (Weiser, 1991). The variability of location context in the forms it exists

[1] Originally, Weiser used the term "ubiquitous computing" in his work. However, as Ye et al. (2008) have noted, the distinction between pervasive and ubiquitous becomes less and less pronounced, and both terms are used interchangeably today.

throughout a multitude of places, situations and technologies nowadays is a hurdle on the road to achieving the everywhere. Lowering and softening this hurdle is the goal of this thesis.

We decompose this high-level goal into a number of smaller, more concrete research challenges, for which we formulate a set of questions, followed by a contribution we offer with respect to each posed question.

Challenge 1: Uncertainty-aware user-friendly location modelling for pervasive environments

An important criterion of a pervasive application's success is seamless and unobtrusive context-aware adaptation. Since location is exposed to high variability in a pervasive environment, a suitable pervasive location model should cope with this variability; that is, it should provide sufficient support for uncertain and incomplete knowledge. On the other hand, such a model should not bring complication. Therefore we have to assure that the frequently changing location information is supported and manipulated easily and is perceived equally well by all parties – the developers, the designers and the users of pervasive applications. Besides, location is hardly ever considered in isolation in a pervasive environment; therefore it is important that a pervasive location model has a means by which it is integrated into the overall context model, so that it can be used together with other contextual information when needed.

Thus, the above discussion can be summarised into the following research question (RQ1):

RQ1: What is a way to approach location and spatial arrangements between interacting resources in a pervasive environment that would:

a) keep the model easily graspable, supported and manipulated at all stages of an application's creation and use?
b) have special attention paid to handling uncertain and incomplete location context?
c) keep location context open to other parts of the environment?

Solutions to represent the locations and relations among entities in a pervasive environment are often developed either application- or environment-specific (e.g., Beigl, Zimmer, and Decker, 2002; Pulkkinen, Bhattacharya, and Nurmi, 2011) or cover RQ1's features partially (e.g., Hu and Lee, 2004; Satoh, 2007; Ye et al., 2007; Glassey, 2009; Stevenson et al., 2010). Therefore we sought to approach and represent the location context in a way that would allow us to incorporate all RQ1's aspects.

Chapter 3 presents a user-centric model of spatial arrangements around a resource in a pervasive environment. The model's main characteristics are the ease of reference between resources and the simplicity of use and support. The model is extended with concepts that handle uncertain and incomplete location information with the help of a set of spatial relationships and measures based on fuzzy logic. The model is represented as an ontology so that it can be merged with other context ontologies. We illustrate its use within a framework for managing pervasive applications by means of extending the framework with a spatial service that implements the support of location context. Some of the results of the investigations that we reported and discussed in Chapter 3 inspired us to inquire into the challenges that follow.

Challenge 2: A unified uncertainty-aware view on location determination

Knowing about location becomes much more beneficial when this information is ubiquitously available. This demand is helped by the rapid development of technology, so that better and improved techniques to determine location appear and the number of places where a user can be located increases. On the other side, the resulting diversity of approaches to location determination calls for additional efforts that will lower (or even completely remove where possible) the barriers between the many and diverse means of location determination. In other words, in order to allow pervasive applications to fully benefit from this ubiquity, not only should a suitable solution make location sensing recognisable and collectable ubiquitously, but also be able to cope with and be robust with respect to the variability and limitations of location sensing capabilities. In turn, the variability and limitations unleash uncertainty, so that additionally, the solution should preserve the uncertain and incomplete information that appears.

There are a number of existing approaches and systems to address multiple location sensing options, focusing on different aspects, such as diversity, uncertainty, scalability, modelling, etc. (e.g., Location Stack (Hightower, Brumitt, and Boriello, 2002); MiddleWhere (Ranganathan et al., 2004); PerPos (Langdal et al., 2010a, 2010b); LOC8 (Stevenson et al., 2010); PIMS (Knauth, Kistler, and Klapproth, 2009); Kurschl et al., 2008; Glassey, 2009; Opperman, 2009). As it is with RQ1, our focus is on making the variability equally viewed at all stages – by the producers of localisation systems and the consumers of location data. Therefore we mainly aim at the approach's practical aspects, such as the easiness of the data representation and manipulation. While preserving and partially re-using the primary major aspects already mentioned among the existing solutions, we are additionally seeking to provide the means and instructions on how to deal with the variability that different localisation systems possess, as well as to make its introduction and support easier. We summed up the above into the following research question (RQ2):

RQ2: How to handle any available location determination option throughout an entire pervasive environment, reflecting the uncertainty and limitations of each option in the location model, in a way that bridges the gap between the providers and the consumers of localisation data?

To answer RQ2, Chapter 4 introduces a unified view on location determination with the help of an ontology of localisation systems that models their metadata and explicitly focuses on the uncertainty, which is intrinsic to sensing but different from system to system. The unification approach is accompanied by a number of software tools and instruments that make the inclusion of a generally unknown arbitrary localisation system into the location context model less cumbersome or technology-specific. The approach was then used as the underlying technical base for a location-aware application that we developed in order to address the challenge that follows.

Challenge 3: Investigating user needs in awareness of the variability of location context: the case of a graphical user interface

Naturally, it is the user satisfaction that must always be placed in the foreground of any application's behaviour. The multitude of situations users can experience throughout the environment necessitates seeking approaches to handling the variability of location context, as well as forms of presenting it, which will accommodate this multitude and will not become a cognitive burden to users.

In general, the area of location-aware adaptation of an application's graphical user interface is considerable, with many aspects of adaptation and user awareness covered elsewhere: a few examples involve the works by Butz et al., 2001; Baudisch and Rosenholtz, 2003; Burigat, Chittaro, and Gabrielli, 2006; Dearman, Inkpen, and Truong, 2010, and many others. However, fewer studies have been devoted to understanding how (much) revealing the variability of location context in the presence of uncertainty affects user experiences (e.g., Dearman et al., 2007; Lemelson, King, and Effelsberg, 2008; Burigat and Chittaro, 2011; Damián-Reyes, Favela, and Contreras-Castillo, 2011; Lim and Dey, 2011). Besides, some of them employed alternative validation approaches, such as paper-based questionnaires (in Lemelson et al., 2008) or Mechanical Turks (in Lim and Dey, 2011), thus making the area of results obtained in real-life settings under-represented. Therefore, speaking about the visual form of awareness, the following question can be asked (RQ3):

RQ3: Which aspects of the variability of location context should be hidden from and revealed to users interacting with the help of a graphical user interface, and in which form?

To answer RQ3, we employed a custom map-based navigation application, using which we performed two real-life experiments in a realistic environment. Chapter 5 describes the application's visualisation strategy and reports on the details and the outcome of the experiments. Based on the analysis of the experiments' flow and results, we proposed a set of design guidelines and implications on visualising some aspects of the variability of location context in pervasive applications and their presentation to users.

Challenge 4: Combining location and social context: the case of a vehicular ad-hoc network (VANET)

We already noted, while introducing Challenge 1, p.4, that the location context is rarely addressed in isolation. Therefore an important part of considering the variability of location context is to make it meaningful within a particular application domain. Here, we mean to let the application exploit this variability in combination with the application's other context in order to provide improved assistance or to assist in a wider range of tasks. Obviously, it is practically impossible to address such a combination in its general form and one has to consider applying the variability of location context to within an area of interest, i.e. where knowing about location is proving beneficial.

While a number of successful examples of considering the two context areas – location and social – together is considerable (e.g., Foursquare[2], CityFlocks (Bilandzic, Foth, and De Luca, 2008); Connecto (Barkhuus et al., 2008); Arminen, 2006; Li and Chen, 2009), fewer attempts in this regard have been made to address transportation systems (e.g., Connected Traveler (Manasseh, Ahern, and Sengupta, 2009)). Therefore we support our final challenge with the following research question (RQ4):

RQ4: How can location and social context be combined in order to improve information filtering in a large-scale vehicular ad-hoc network (VANET)?

[2] http://foursquare.com

Chapter 6 presents our investigation of a ubiquitous help-seeking/provision scenario in the area of large-scale vehicular ad-hoc networks (VANETs). We introduce an approach, in which information about vehicles' locations and locomotion is intertwined with drivers' social profiles in order to offer a more efficient message communication scheme that delivers more relevant information. We evaluated the performance of the approach using a real-time discrete event-based network simulator that ran on a large-scale vehicular network using a realistic dataset.

1.3. Topics not included

Location is a truly multifaceted concept. There are many location models and representations that have been proposed and developed, as well as aspects considered, in order to fulfil yet a wider range of goals, aims, and scenarios of user activities and application purposes. As a result, there is a plethora of separate branches and subtleties identified in location context that are attracting researchers' and practitioners' dedicated attention. This section outlines some popular domains and topics of research on location context and its applications that are addressed elsewhere and that this thesis does not focus on.

We do not deal with algorithms or techniques for improving the quality or the reliability of location data. Such areas as novel approaches to object localisation, their error and proximity estimation, or sensor fusion strategies are just a few of the many examples. These are separate areas of research and a lot of efforts have been, and are still being, done in this regard (e.g., Lemelson et al., 2009; Nakamura and Shinoda, 2010; Matic et al., 2010; Backstrom, Sun, and Marlow, 2010; Widyawan et al., 2011). We rather consider the very location information we currently have at hand and investigate what can, and should, be done with this information so as to make the most use out of it in the given conditions.

Next, we do not directly address location privacy, security or trust aspects. There are many examples of dedicated work in these and other related areas (e.g., Ardagna et al., 2007; Bernheim Brush, Krumm, and Scott, 2010; Boesen, Rode, and Mancini, 2010; Scipioni and Langheinrich, 2010; Tang, Hong, and Siewiorek, 2011; Peddinti, Dsouza, and Saxena, 2011). We suggest that this thesis' findings be superimposed with existing results in the areas of hiding, protecting, requesting, or sharing location information.

Finally, we do not address time- or life-critical scenarios, such as emergency environments, where the cost of a failure to provide an immediate response or the cost of reporting a false or imprecise position is too high. These environments usually require a special treatment with different priorities (Fischer and Gellersen, 2010; Fuchs et al., 2011) whereas we rather address applications for everyday common use by ordinary users.

1.4. Thesis structure

This thesis is organised as follows:

Part I is devoted to modelling the location context and its variability throughout a pervasive environment. In Chapter 2, we begin with an overview of existing approaches to location representation and discuss existing models to describe the location context in pervasive environments. We identify variability issues resulting from this modelling and representation, from which we derive a set of requirements for a pervasive location model. We then review existing ways and technologies for location sensing, outline their limitations and issues, identify sources of variability in location sensing, and conclude with a set of requirements for pervasive localisation modelling. Chapter 3 presents a resource-oriented (ego-centric) model of the spatial relations between resources in a pervasive environment and ontology support to work with it. The model reflects the variability of location data, and the applicability of the model is illustrated in a set of simulated use-cases on user interface distribution scenarios as part of an ontology-based framework for coping with a pervasive environment. Chapter 4 introduces a model to describe location sensing and determination capabilities, representing each in a unified format and using a number of tools and corresponding guidelines to working with it and to making an arbitrary location provider part of the processing framework and the environment.

Part II illustrates where and how the variability of location context can be used. Chapter 5 presents the results of two experiments in which we employed a map-based navigation application to help us investigate user needs in awareness of their location tracking conditions. Based on the analysis of the experiments' flow and results, we proposed a set of guidelines and implications on visualising some aspects of the variability of location context when developing location-aware pervasive applications. Chapter 6 shows how location context can be combined with other information to assist in a ubiquitous help system in a help-seeking scenario in the area of large-scale vehicular ad-hoc networks (VANETs). Namely, we interweaved information about vehicles' locations and locomotion with drivers' social profiles in order to improve information filtering in the communication between interested parties.

Finally, Chapter 7 summarises thesis' findings, draws overall conclusions and outlines some directions for further improvements.

Part I

Modelling aspects

Chapter 2

Background and requirements

The chapter consists of two logical blocks, each in turn comprising two sections (2.1-2.2 and 2.3-2.4, respectively). Section 2.1 covers the state of the art and common views on location modelling, discusses open issues and existing approaches and solutions therein. Following, section 2.2 concludes the first block with a number of requirements that the location context – a crucial component of a pervasive environment – is expected to meet in order to become an efficient and effective part of the overall pervasive context model. In the second block, section 2.3 provides an overview of possibilities to obtain location information throughout the environment and discusses existing models and systems for handling and manipulating with this information. Section 2.4 then presents several requirements that aim to ensure the interpretation and proper handling of any location provider in the environment.

2.1. Location modelling

2.1.1. Representation

Dey and Abowd (2000), in their definition of context, use the term "entity" as a common name to refer to concepts that are considered relevant. Location context supplies us with information provided in response to a request about an entity's whereabouts, or simply to the question "Where is entity N?". This question can be answered in a diversity of ways.

Figure 2.1 illustrates a taxonomy for addressing location in a pervasive environment proposed by Dobson (2005). Using this taxonomy, one can refer to an entity's location in any of the following ways:

- using an absolute position (e.g., 20N, 20W);
- specifying a name from some agreed namespace (e.g., room 0.05A) or category (e.g., conference room);
- specifying a space related to this entity (e.g., in his office, in his car), to some other entity (e.g., at John's home), or to some other entity's location (e.g., with John);

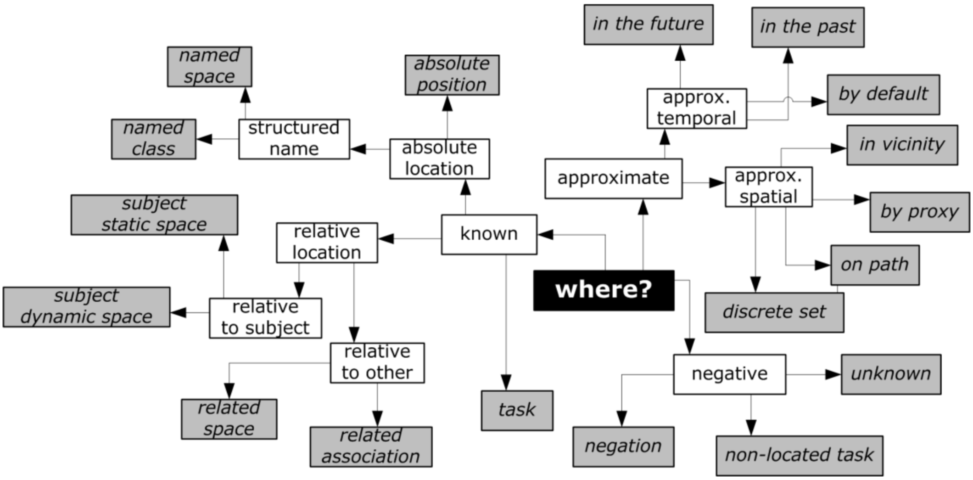


Figure 2.1. A taxonomy of location in pervasive computing, adapted from Dobson (2005).

- mentioning a moment in the past (e.g., at 7pm yesterday) or in the future (e.g., at 7pm tomorrow);
- specifying a vicinity (e.g., within 2km) or elongated region (e.g., between A and B);
- specifying some discrete set (e.g., either in room A or B or C);
- using a proxy (e.g., a mobile phone);
- linking the location with a task (e.g., visiting the headquarters) or with some known regular activity (e.g., at a weekly Monday planning meeting);
- giving a negative answer (e.g., not here) or no answer at all (e.g., no idea);
- giving an answer without any location (e.g., on holiday).

As we see, a range of possible answers to the same single question is indeed diverse, with each answer relying on its own representation of the spatial terms. However, it is quite improbable that one application will require all of them simultaneously. Vice versa, different applications would need different representations that would be compliant with the application purposes. In this regard, the two main questions that appear here are 1) *what information is necessary*, and 2) *how to represent what has been identified as necessary* (Glassey, 2009). Several approaches to representing and describing this information, i.e. the spatial world, are possible.

Quantitative, or geometric, descriptions usually refer to objects with the help of a coordinate system and a frame of reference. Probably, the most famous example is the GPS coordinates that have a universal frame of reference and therefore are always universally translated. Ultimately, the main quantitative measures in such systems are distances, so that an advantage of quantitative systems is their ability to determine how far objects are from each other. However, such a language is not always straightforward to work with (e.g., where is "20N, 20W"?). Besides, numbers alone may not be sufficient: for example, a close in the quantitative terms object may turn out to be inaccessible since it would simply be located behind the wall, e.g. in the adjacent room.

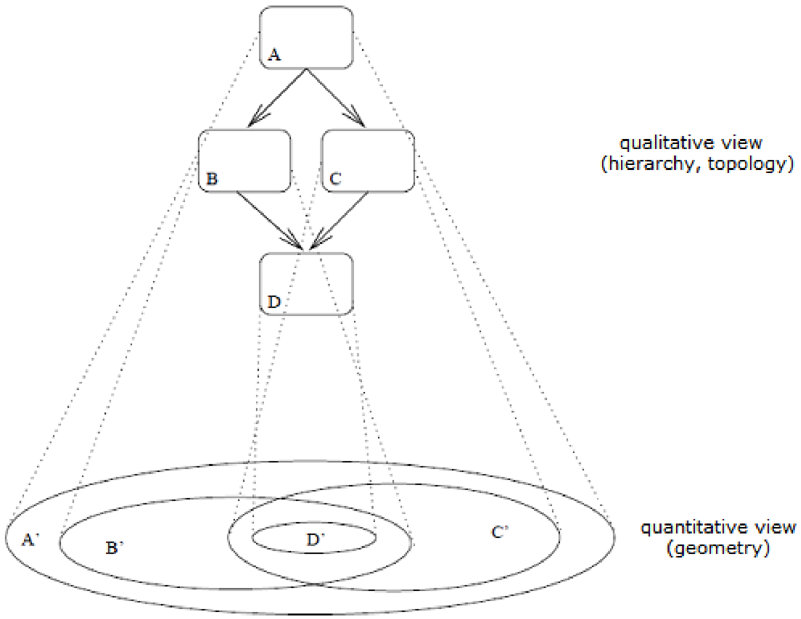


Figure 2.2. The qualitative and the quantitative approaches to location representation, combined together, form the semi-symbolic, or hybrid, model. The graphics reproduced from (Leonhardt, 1998).

Contrariwise, a qualitative, or symbolic, description refers to an entity's location using symbols or correlations with locations of other entities. A typical illustration would be to say that an entity is, for example, "in Room B". An advantage of this type of reference is that locations are usually expressed in natural language, so that understanding and reading them is easy. Usually, qualitative descriptions are organised in a way that would form containment and connectedness relationships between entities, such as "*on* floor 2 *in* wing C *in* building D". However, the underlying models are usually built application or domain specific, and so is the corresponding symbolic mapping. For example, what is called "room" in one representation may be called "office" in another. Besides, qualitative models often provide only coarse levels of granularity, so that the model only allows us to say that two objects are, for example, in the same building or a room of a building, but nothing beyond. That is, using just qualitative descriptions without additional knowledge, we are unable to answer questions like "Which printer in building A is closer?"

The advantages of the qualitative and quantitative approaches tied together form the semi-symbolic, or hybrid, representation, devised originally by Leonhardt (1998) and further elaborated, using similar concepts, by Jiang and Steenkiste (2002). This approach enriches the qualitative representation of spaces with a quantitative view, introducing a frame of reference and the coordinates applicable to each space within the qualitative model (see Figure 2.2).

As Dobson's (2005) taxonomy shows, the interpretations of the qualitative, quantitative, or hybrid representations are rather flexible, so that a suitable interpretation is chosen in accordance with custom requirements. Practical illustrations of approaches that use either representation, or their combination in either form dictated by the context of use, are many; some particular examples of models and their applications included Cohn et al.'s (1997) region connection calculus (RCC), Lorenz et al.'s (2006) hybrid spatial model for indoor environments, Satoh's (2007) M-Spaces model for pervasive environments, Ye et al.'s (2007) unified space model for complex environments.

2.1.2. Realisation and management

As such, a representation of location concepts alone covers only a single aspect.

Firstly, location in ubiquitous computing is rarely considered in isolation. We refer to the hour of the day, the type of mobile device, the active task, etc. Here, location is often seen as an empowering force for various other situations and subsequent context usage, such as the identification of places (e.g., Koile et al., 2003; Nurmi and Bhattacharya, 2008), place-based activities (e.g., Dearman and Truong, 2010), availability (e.g., Lovett et al., 2010), situational awareness (e.g., Streefkerk et al., 2008), etc., so that location receives additional semantics with which it becomes associated.

Secondly, before any such action can be performed, one needs to make the location model practically useful for a computing environment, i.e. to find an approach to realising and organising the knowledge about the location model for its further operation, manipulation, or linkage. In this regard, Nexus (Hohl et al., 1999) was an early attempt to build a common platform that location-aware applications could use, making the communication between the existing applications and the creation of new ones easier. It proposed a common augmented world model representing static real-world structures (buildings, streets), mobile real-world objects (humans, cars) and virtual objects augmenting the world, each distributed among spatial model servers responsible for their own region. But with the shift to more mobile and ad-hoc nature of applications, it became less suitable and applicable in unfamiliar environments. Overall, Table 2.1 shows the results of Strang and Linnhoff-Popien's (2004) comparative analysis of the appropriateness of several existing approaches to context modelling for the case of pervasive computing environments. Using a set of requirements, they concluded that ontology-based models proved to be the most suitable and promising means.

Table 2.1. Appropriateness indication of approaches to context modelling, based on a set of requirements(*) with respect to pervasive computing. Reproduced from (Strang and Linnhoff-Popien, 2004).

	dc	pv	qua	inc	for	app
Key-Value Models	-	-	–	–	–	+
Markup Scheme Models	+	++	-	-	+	++
Graphical Models	–	-	+	-	+	+
Object Oriented Models	++	+	+	+	+	+
Logic Based Models	++	-	-	-	++	–
Ontology Based Models	++	++	+	+	++	+

(*) dc – distributed composition; pv – partial validation; qua – richness and quality of information; inc – incompleteness and ambiguity; for – level of formality; app – applicability to existing environments.

Although a number of ontologies for modelling context have been proposed (e.g., Chen et al., 2004; Gu, Pung, and Zhang, 2004; Preuveneers et al., 2004; Ranganathan et al., 2004b), Ye et al.'s (2007b) detailed analysis of a number of most popular then existing ontologies concluded that despite that each of them elaborated on important facets, no single ontology was rich enough to cover pervasive systems sufficiently. These conclusions were further on supported, for example, by Glassey (2009), as well as were followed by the justified appearance of other (location) context ontologies for pervasive computing systems and applications (e.g., Niu and Kay, 2008b; Stevenson et al., 2009; Vanderhulst, 2010; Strobbe et al., 2012). So that ultimately to date, there is still a lack of a single ontology of location context that is approved by the location-aware community.

On the other side, Glassey (2009) showed that despite the many solutions covering a variety of situations, areas, and application domains, the ubiquity of location-awareness was still facing the problem of contention between location modelling and management. In his PhD thesis, Glassey analysed in detail and then explicitly targeted the contention problem. He proposed a Ubiquitous Modelling Platform (UMP) consisting of two complementary independent components for location modelling and management, respectively. The application of that platform demonstrated the possibility to support ubiquitous location-awareness without compromising the location modelling and management concerns.

While our work also falls into the category of providing ubiquitous location-awareness, there are a few differences. Firstly, while Glassey explicitly addresses and analyses the details and issues in stable and reliable communication of the location modelling and management components on the systems level, we deal with the management aspect only partially, employing, where possible, suitable existing solutions such as W2P (see section 4.2, p.65) and GSN (see section 4.3.1). Secondly, Glassey's Relational Location Model (RLM) rather provides the abstraction in which different location models could be evaluated, so that with no core types or relations specified, the details of variability and uncertainty over different location models and tracking were thus also omitted. Different to this, our main focus has been on representing and handling the variability that comes within different location context models, with which we aimed at the representation's practical aspects by considering the developers, designers, and the end users aspect all together. Thirdly, while the evaluation framework for UMP/RLM was based on analytical comparisons of representations and assessments of their flexibility, we analysed the influence of the variability of location context presented during tasks in real-life settings.

2.1.3. Uncertainty

As noted in the previous section, an important characteristic of context is that it is usually prone to errors and uncertainty (Bettini et al., 2010). Damián-Reyes, Favela, and Contreras-Castillo (2011) distinguish between three different forms that the uncertainty in context can take:

- *uncertain context*, i.e. when contextual information is unreliable, weakly controlled, or it is causing doubts in the users' impressions about an application's validity,
- *ambiguous context*, i.e. when it is impossible to distinguish or interpret the contextual information,
- *wrong context*, i.e. when the received information is incorrect or irrelevant (e.g., outdated),

which, as Ye (2009) explains, can be dealt with in two ways, qualitative and quantitative.

The quality of context information is the extent to which this context information corresponds to the real world (Castro and Muntz, 2000). Attributes of the quality of spatial context information involve (e.g., Gray and Salber, 2001; Lei et al., 2002; Judd and Steenkiste, 2003), for example, *coverage*, i.e. the amount of contextual information that can be obtained; *resolution*, i.e. the smallest perceivable element; *accuracy*, i.e. the range in terms of a measure of the property; *repeatability*, i.e. a measure's stability; *frequency*, i.e. the sample rate, the temporal equivalent of resolution; *freshness,* i.e. the age of the reported information.

The quantitative approach involves such methods as fuzzy logic (Zadeh, 1965), probabilities, and their variations and enhancements such as Bayesian Networks (Heckerman, 1996) or Dempster-Shafer theory (Sentz and Ferson, 2002), to help with processing imprecise and incomplete knowledge. With the appearance of more complex systems, this approach of dealing with uncertainty aspects in context-aware systems has gained more popularity (Ye, 2009), and a number of approaches have been proposed in this regard (e.g., Ranganathan et al., 2004c; Abdelsalam and Ebrahim, 2004; Anagnostopoulos et al., 2005; Ye et al., 2008b). In particular, attempts have been made to incorporate the probabilistic and fuzzy-based uncertainty support into ontologies (Ding and Peng, 2004; Gu et al., 2004b; Truong, Lee, and Lee, 2005; Gu et al., 2007).

Each approach is able to handle uncertainty to some degree, performing well in some aspects and not so well in others, so that no universal solution can be provided. Similarly to the main location modelling, to date there is no single approach to addressing uncertainty within a context model (and its ontology), and the application of a particular approach still depends on the nature of uncertainty (Ye, 2009).

2.1.4. Discussion

The great success and the rapid growth of the number of activities one can do with the help of location-based pervasive applications and services today has resulted in new challenges and domains of applicability of location context (Dey et al., 2010). In a pervasive environment, where the physical and the digital worlds connect, things complicate by the fact that the heterogeneity of such an environment dictates its own rules for the spatial world to be an efficient part of the environment. The diversity of tasks and scenarios, together with a lack of recognised and universally approved standards, has made it uneasy to agree on what spatial context to include. The corresponding model can neither be built heavily task-dependent, nor become too specific or too complex to be included. It is still often unclear how spatial information should integrate with existing models. To aid in this, the next section presents a number of requirements we have identified that the location context within a pervasive context model is expected to meet.

2.2. Requirements for pervasive location modelling

As stated, it is often simply impossible to perform an extensive calculation or operate a detailed and powerful, but complicated description of location data, which cannot be straightforwardly understood and incorporated into a pervasive application. Yet such information should at the same time provide a rich enough technique to cover the many varying scenarios, in which the knowledge about location is necessary. It has to be equally available and realisable by the creators as well as graspable by and intuitive to users of these applications. With these considerations in mind, we now introduce a set of requirements a location model suitable for a pervasive environment should comply with:

- *be reasonably simple*. Simplicity is a crucial factor in dealing with the high dynamics of a pervasive environment. Often, there will be no time, or space, or resources – or simply no need – to take into account each and every detail of a comprehensive but computationally expensive model. The high dynamics may simply request to switch to a neighbouring situation, while the one in question has not been completed. Alternatively, not all parts of the environment will have enough resources or power for providing the expected information.
- *be human friendly*. We need an easily recognisable interpretation of the location information by both the users, who have become a considerable part of pervasive environments, on the one side, and the designers and developers, who rely on the proposed model at the application creation stage, on the other. An otherwise simple model that operates non-straightforward concepts is not straightforward to handle either and therefore will be shown reluctance.
- *be able to handle uncertain and incomplete knowledge*. Uncertainty in location context is inevitable; handling it is paramount (Bettini et al., 2010). Therefore the application creators have to have an ability to work with and manipulate uncertain location context in accordance with the application needs, for the amount of such context is application specific and often depends on third-party factors. This necessity to be able to take into account factors coming from other context raises our next requirement that suggests that a pervasive location model is expected to
- *have an open handling scheme*, so that information about location context can be integrated into a bigger context model and thus extended, when

necessary, to a whole range of pervasive applications. In other words, the model should be built mergeable with, integratable with, and recognisable by other contexts.

In general, the question of addressing location context is not bounded by these only requirements, and ultimately, one can suggest other aspects that the location model within a pervasive application would be expected to follow. In this regard, our requirements' main focus has been on the location model's practical considerations and runtime support during all stages of a pervasive application's creation and usage.

But prior to any operations on location context, which assign it with a meaning and allow applications to use it, this context has to be produced. The next section gives an overview of the techniques, methodologies and systems used to generate, obtain and manage the location context production; that is, talks about location sensing.

2.3. Location sensing modelling

2.3.1. Sensing diversity

The Global Positioning System (GPS) (Hoffmann-Wellenhof et al., 1992) is, without exaggeration, the most famous and widespread technique for locating objects. In GPS, an object's location is determined using a number of time signals received from the GPS satellites orbiting around the Earth. There are thirty one active satellites in total, of which about nine are visible at any point at any moment. Each satellite broadcasts messages that contain its precise orbital position and the time of sending. Using this information, a GPS receiver calculates the distance to each satellite and thus determines a sphere, centred at the satellite, of possible locations at the computed distance from the satellite. By intersecting the spheres from all satellites visible at the moment of calculation, the receiver computes its own location. Minimum four satellites are required for a successful location fix. The accuracy of a GPS location thus depends on the number of visible satellites and can vary from a few to sometimes around a hundred metres. The main drawback of the GPS technology is that it requires a direct line of sight between the satellites and the receiver for the signals to be received. This makes GPS unusable indoors and in urban areas where the materials of buildings and other constructions distort or block satellites' signals. Although attempts are made to investigate the performance of GPS indoors (e.g., Kjærgaard et al., 2010), the fact that most people's every day activities take place indoors or in busy city areas makes GPS unsuitable for pervasive environments.

Another popular method of localisation uses WiFi access points (APs) as location signals. A well-known approach to wifi-based localisation is called "fingerprinting" (Bahl and Padmanabhan, 2000). At the core of this approach is a collection of WiFi scan samples, called fingerprints, recorded at a number of known spots, whose locations are pre-calculated somehow else[3]. At each of these spots, all available WiFi APs are searched, creating this spot's unique "fingerprint" in terms of the APs' received signal strength indicators (RSSI). With this fingerprint map at hand, the device being located compares the received fingerprint of its current location in order to find the closest fingerprint in the fingerprint map, or applies a location determination algorithm (such as triangulation) on the base of known locations of the closest fingerprints. There have been many efforts to exploit WiFi in localisation; one famous example being the PlaceLab project (LaMarca et al., 2005), with its practical applications met elsewhere (e.g., Cheng et al., 2005; Hightower, LaMarca, and Smith, 2006).

A similar idea of using signals as location indicators is utilised in GSM-based localisation where the known locations of GSM-based stations are used as reference points and fingerprint sources (e.g., Varshavsky et al., 2007). Another example of the application of the fingerprint approach is PowerLine Positioning (Patel et al., 2006). Instead of wifi, this location system's fingerprints are created by using the residential, and later also commercial (Stuntebeck et al., 2008), powerline as the signalling sources.

Other techniques have also been used to locate objects in smaller environments, where better accuracy or reliability is required. For example, in the Active Badge system (Want et al., 1992), one of the earliest systems for indoor localisation, small devices called "active badges" were attached to objects (mainly, personnel) being located, and transmitted unique infrared signals. The network sensors then received the signals, and the location of the active badge was determined using the information from the received signals, providing room-level accuracy. The Active Bat system (Harter et al., 2002) is based on a grid of ultrasonic receivers attached to the ceiling of a building which receive signals emitted by active bats. The system is able to locate objects within around 10cm 95 percent of the time; however, due to its extended infrastructure, this system is not easily deployable.

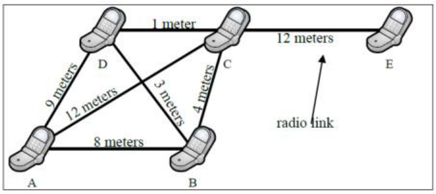
Ubisense[4] is a solution for real-time precise tracking based on the ultra-wide band (UWB) radio technology. A typical Ubisense setup consists of a set of firmly fixed (ubi)sensors that track locations of ubi(tags) throughout a certain area defined by the position and orientation of the sensors. The advertised precision of

[3] Not necessarily of the GPS origin; any custom frame of reference can be used.

[4] http://www.ubisense.net

tracking with Ubisense is 15cm in 3D space; however, when tagging humans, the performance usually varies depending on the direction and orientation of the body and the visibility of a (ubi)tag by the (ubi)sensors (Coyle et al., 2007). Although Ubisense is quite expensive and is therefore mainly affordable by businesses or organisations, which limits its ubiquitous applicability, it remains an important player.

So far, the described approaches have, in either form, employed an infrastructure that made location sensing possible. An alternative approach to infrastructure-based localisation is peer-based localisation. As it follows from the term "peer-based", this method of localisation is not standalone and is a result of collaboration of a number of resources, so that the locations of devices are determined relative to each other (e.g., see Figure 2.3). It is primarily used in situations where a possibly short term but immediate and often ad-hoc localisation is needed. A few examples involve the RELATE ultrasonic system (Gellersen et al., 2010), the Virtual Compass, which is a peer localisation system based on radio signals (Banerjee et al., 2010), or an opportunistic pedestrian localisation system exploiting the features of the Bluetooth communication (Wagner and Kray, 2010).



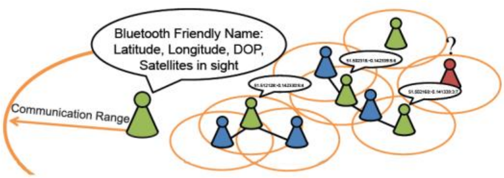


Figure 2.3. In peer-to-peer localisation, devices directly communicate with co-located peers and exchange the information that is required by the localisation algorithm. For example, the Virtual Compass approach uses received signal strength indications (RSSI) for distance calculations (left), and Wagner and Kray's opportunistic approach uses Bluetooth-friendly names encoding GPS positions (right), so that localisation becomes possible without the need for an actually functioning communication channel. Graphics reproduced from (Banerjee et al., 2010; Wagner and Kray, 2010).

Furthermore, there are also approaches and techniques, which explore yet other concepts and attributes as location information sources: e.g., FM signals (Popleteev, 2011), a mobile phone camera (Ravi et al., 2006), or a combination of optical, acoustic, and motion attributes as an ambient fingerprint (Azizyan et al., 2009). So that all together, the variability of the location data produced and the parameters involved into each particular localisation is yet growing.

2.3.2. Processing and managing

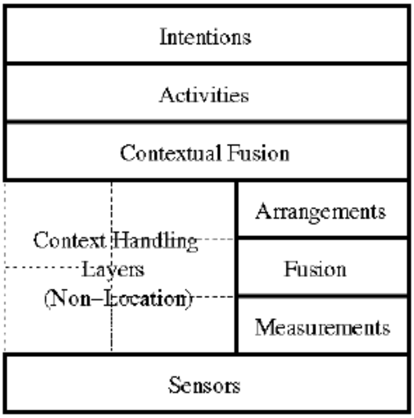
We have seen in the previous section that the problem of location determination is presented and studied extensively. However still, it is not ultimately solved, so that attempts are made to create new approaches and technologies, as well as to improve existing ones. In facing this multitude, an important aspect towards the ubiquity of location-awareness is the need to have a means to consider the diverse location determination technologies simultaneously, as well as to manage them.

One way of overcoming drawbacks and limitations of separate independent technologies is to consider the location data provided by different localisation systems together so as to produce another result improved in some way (e.g., more accurate). This process is called "sensor fusion" (Hightower and Boriello, 2001). Some examples of its application involve combining a wifi-based and an acoustic localisation (Hii and Zaslavsky, 2005), merging Bluetooth and WLAN technologies (Aparicio et al., 2008), using RF and ultrasound beacons in unknown environments without any map information at all, thereby making it suitable for emergency applications (Fuchs et al., 2011).

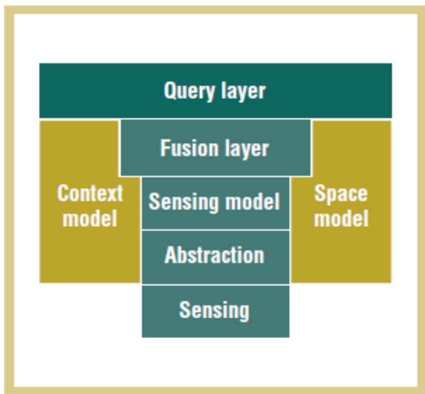
Apart from fusion, indirect operations and manipulations with sensed location data may be performed. For example, the Location Stack (Hightower, Brumitt, and Boriello, 2002) was an early example of a system that had fusion as part of a more sophisticated approach to dealing with the many location sensing possibilities. It also included a layer for considering the location context's impact on other context, i.e. options already mentioned earlier in section 0. Another example is to use the location context at hand for deriving new context and treating problematic situations, such as solving location conflicts coming from granularity variance, false positive or false negative results (e.g., Niu and Kay, 2008a), or providing a correct handover between several tracking opportunities, thereby extending a location-aware application's usage area (e.g., Hansen et al., 2009).

Therefore, similarly to managing location models, a system is needed that would help organise and structure the acquired sensed data for its further manipulation. In this regard, such systems as the already mentioned Location Stack (Hightower, Brumitt, and Boriello, 2002), MiddleWhere (Ranganathan et al., 2004a), or PoSIM (Bellavista, Corradi, and Gianelli, 2008) are earlier examples of middleware that support introducing new location sensing providers and provide a framework for data processing (e.g., Bayes-based probabilistic reasoning to fuse multiple sensor readings within MiddleWhere). But as Langdal et al. (2010b) noted, the systems have limitations with respect to the variability of features in location information that can be included.

LOC8 (Stevenson et al., 2010) is an ontology-based programming framework – with an architecture partly based on the Location Stack model (see Figure 2.4) – for supporting the use of location context by pervasive applications' developers. Using the two ontologies – an ontology of spaces and relationships between them (the space model), and an ontology describing multiple location providers and their metadata in order to incorporate positioning uncertainty, such as the accuracy of measurements and coverage area (the sensing model) – the framework allows developers to take care of the relationships between interacting resources so as to reason about location context. A similar example of ontology-based location support and reasoning is a Semantic Location Service proposed by Coronato, Esposito, and Pietro (2009). It is based on two ontologies that describe, respectively, entities of a pervasive environment (such as a user, device, or positioning system) and semantic locations (such as a room or corridor). Although the model does not involve fine details of the very sensing process such as a localisation system's update rate, the application of the approach illustrates the potential of modelling with ontologies in situations where multiple localisation systems are to be supported.



The seven-layer Location Stack



LOC8's layered architecture

Figure 2.4. LOC8's architecture is based on the Location Stack model but differs in a number of respects, such as focusing on a different measurement level abstraction, separating the space model from the sensing model, taking a cross-layered approach to context fusion, as well as realising a different fusion method. Graphics reproduced from (Hightower et al., 2002; Stevenson et al., 2010).

PerPos (Langdal et al., 2010b) is a recent example of positioning middleware that advocates so called "seamful" design, in which low-level details of location sensing data are made available within the middleware, so as to let one, when required, look inside the middleware and focus on and reveal specific seams. In particular, uncertainty of the positioning process is managed within this middleware as an independent concept (Langdal et al., 2010a), as shown in Figure 2.5. Although, according to Langdal and the co-authors, the initial idea of the seamful design was intended towards end users, the PerPos middleware supports developers, so that the imperfections of the many sensing technologies throughout a pervasive environment could be addressed during the application's development phase.

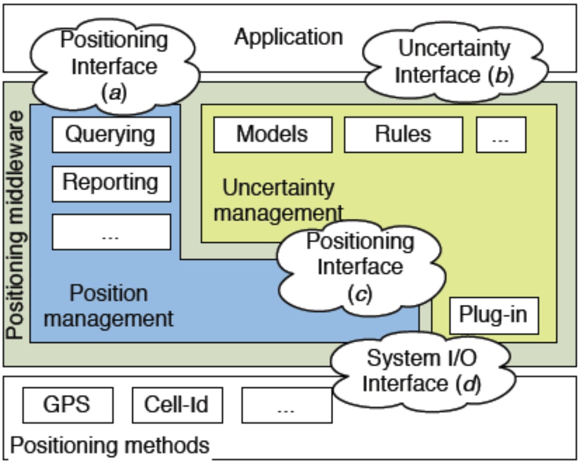


Figure 2.5. An architecture for an uncertainty-aware positioning middleware, in which position information and uncertainty are both considered and managed as first class entities. Graphics reproduced from (Langdal et al., 2010a).

There are also some examples of attempts towards systems for managing the location sensing process in rather domain-specific environments, such as PIMS (Knauth, Kistler, and Klapproth, 2009), a position management system designed to support ambient assisted living scenarios, or Kruschl et al.'s (2008) system for managing multiple localisation options, which is used in improving occupational safety in industrial plants.

2.3.3. Discussion

The diversity of location determination approaches generating highly variable location data, the multitude of situations in which knowing about location is necessary or at least beneficial, the heterogeneous nature of the environment – these and further similar factors considerably complicate location support for pervasive applications. As a result, there is no universal support and management, and each existing system covers one particular aspect of this heterogeneity, while doing less so towards the others. In this regard, in order to make the aspect of location-awareness and support more ubiquitous, we suggest that the underlying system for dealing with location sensing should mainly ease the use of an arbitrary location provider available in the environment at a given moment and handle the intrinsic, i.e. initial, variability properly. The next section presents several requirements we have elicited that such a system is expected to follow.

2.4. Requirements for pervasive location sensing modelling

As stated, our main aim in elaborating on the details of the acquired location data is to ensure the interpretation of any location provider, originating from either an infrastructure- or peer-based localisation, either geometric or symbolic data, accurate or coarse-grained, and at any place and time. These considerations converge on the following three features that the system for working with pervasive location sensing has to provide with respect to the location sensing data:

- *availability and scalability.* Wherever the environment is able to generate location measurements, they have to be understood by the system and further delivered to the application. This requirement makes us pay attention to two aspects: the format and the structure of the location data and the means of its reception.
- *flexibility.* The provision of this feature ensures that the system is prepared for the variability of location information, as well as for changes in the environment.
- *non-centralised processing.* The approach to location data processing must support independent manipulation of the information obtained locally; at the same time it must have connections to and be able to receive and work with remote data.

Here, two important clarifications have to be made with respect to the suggested requirements. Firstly, these requirements do not claim exhaustiveness; we do not aim to cover each and every aspect of pervasive location sensing but are rather seeking to make a location provider at hand, i.e. available at a given moment, an equal player, as well as to treat the absence of any location provider appropriately. Connected to this, secondly, the requirements do not make any assumptions on which location information arrives, i.e. we do not focus on a location model or a coordinate system behind the received data. As we have seen, there has already been created support in this regard, so that solutions such as LOC8 or RLM (see section 2.1.2) can be involved to provide the corresponding abstractions and support. Our primary aim has been to let the results of the heterogeneous location sensing process in a pervasive environment become more ubiquitous by coping with its uncertainty and variability aspects from both the producers' and the consumers' perspective.

2.5. Summary

This chapter has discussed the two main areas of location context – location modelling and location sensing modelling. We have given an overview of the main approaches used to building these models and discussed existing techniques and strategies to manipulate and work with them. Each discussion was followed by a set of requirements that the models are expected to meet in order to succeed in a pervasive environment.

In the next two chapters, we present and discuss the arrangements we have identified on the basis of the specified requirements.

Chapter 3

Arrangements for pervasive location modelling

Parts of this chapter have been published in the following workshop and conference proceedings:

- Aksenov, P., Luyten, K., and Coninx, K., 2008. Reasoning over spatial relations for context-aware distributed user interfaces. In *Proceedings of the 5th International Workshop on Modeling and Reasoning in Context*, pages 37–50.
- Aksenov, P., Vanderhulst, G., Luyten, K., and Coninx, K., 2009. Ambient compass: one approach to model spatial relations. In *Proceedings of the 13th International Conference on Human-Computer Interaction*, pages 183–191.
- Aksenov, P., Luyten, K., and Coninx, K., 2009. Coping with variability of location sensing in largescale ubicomp environments. In *Proceedings of the 1st International Workshop on Sensing and Acting in Ubiquitous Environments*, pages 1–5.

In this chapter, we present a resource-oriented (ego-centric) model of the spatial relations between resources in a pervasive environment and ontology support to work with it. The model reflects the variability of location data, and the applicability of the model is illustrated in a set of simulated use-cases on user interface distribution scenarios as part of an ontology-based framework for coping with a pervasive environment.

3.1. Connecting resources

A pervasive environment is populated with heterogeneous devices and equipment, installed and operating at public places (e.g., modern hospitals) and private locations (e.g., smart homes). All these resources interact with each other, serving the needs of the users of this environment. Regardless of their roles, each device, appliance, or its user are, at any given moment, in the first place at a location. The characteristics of a particular location affect a resource's interaction range, i.e. a proximity within which it can act or be used. The co-existence of several resources within the same area matters to their cooperation and collaboration capabilities. Human visitors come and go, carrying their mobile devices with them; stationary resources get moved, rearranged, and replaced. As a result, the spatial dependencies existing between interacting resources also change: distances vary, directions turn, removed resources may no longer be used as references, etc. It, in turn, affects the interaction details, so that they must be updated and reorganised to reflect the new status.

3.1.1. Motivating scenario

Consider an image viewing application with an additional option to answer a question about the displayed image. It is possible to switch to the next or the previous image in the collection, zoom in and out the currently displayed image, and type answers and/or comments to the question associated with (a part of) the displayed image, in a dedicated text area. Thus, the application's user interface, illustrated in Figure 3.1, contains four functionalities – displaying images, switching, zooming, and typing comments – which can be shared between several resources. Consider also that several resources are currently active in the room: a big screen for public use, a laptop, and two smartphones, all connected to a local wireless network. Initially, the touch-based switching functionality of the image viewer is controlled on one smartphone, zooming is controlled on the other, and the laptop user adds comments. The currently chosen image is shown on the big screen, so that everyone can see it. This situation is shown in Figure 3.2.

While the screen is always at the same location and is turned on, the other devices are mobile. If any of them leaves the interaction space, the complete functionality of the user interface must still preserve, so that the corresponding part of the interface migrates[5] to one of the remaining resources or to a newly arrived resource if one becomes available. In order to be able to recognise such situations and intentions, we need to know about the spatial arrangements between resources at a given moment. We must approach this information in a way that would allow us to derive new facts. That is, we should be able to reason over spatial relations that exist between the resources interacting with each other in the environment, so that we are able to identify that an action is required at run-time.

[5] A user interface is migratable if it can be transferred from one device to another, such as from a stationary PC to a smartphone (Grolaux et al., 2004).

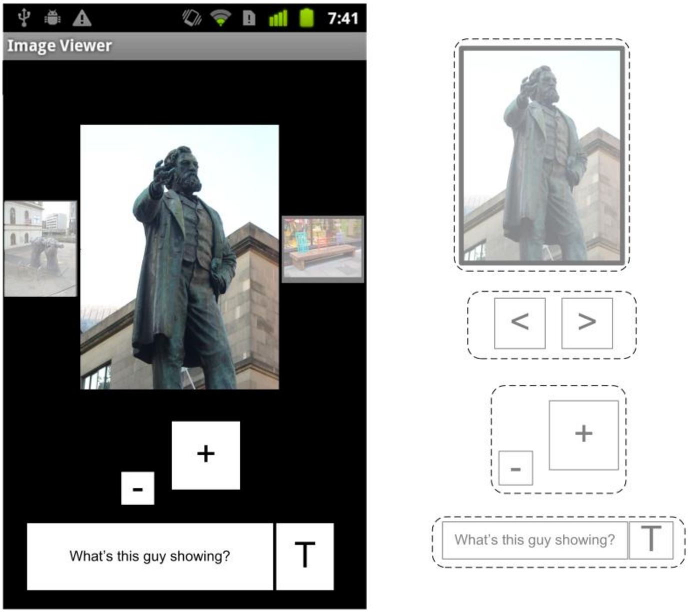


Figure 3.1. An image viewing application's user interface has four parts: viewing, switching, zooming, and typing.

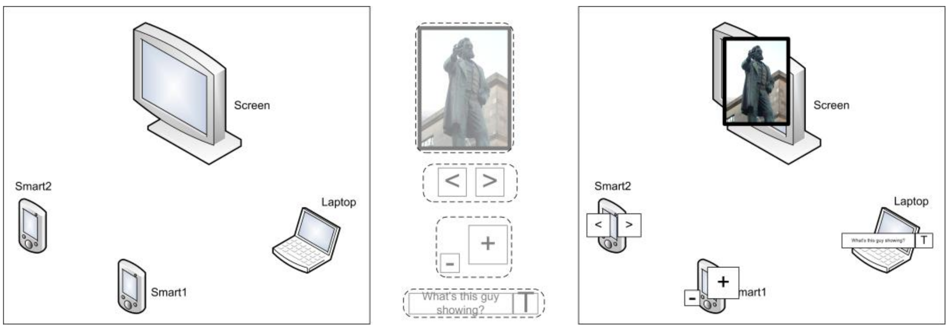


Figure 3.2. The user interface of a collaborative image viewer is shared between four resources with different capabilities most suitable for the corresponding interface functionality.

3.1.2. Graph representation

The arrangement of the mobile resources in Figure 3.2 is flexible: they can freely move throughout the area. If, for example, we now calculate the distances between these resources and attach the calculated values to the corresponding pair of resources, the resulting structure, depicted in Figure 3.3, resembles a graph.

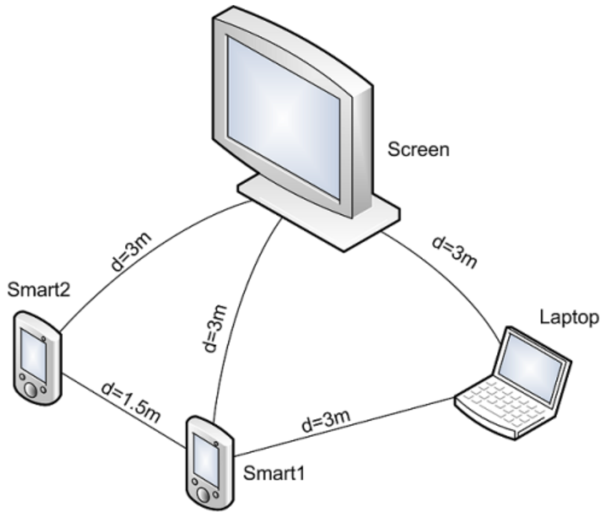


Figure 3.3. A spatial graph modelling a pervasive environment. Resources are the graph's nodes, spatial relationships between resources are its edges.

Formally, Diestel (2010) defines a graph as

> "...a pair $G=(V,E)$ of sets such that $E \subseteq [V]^2$; thus, the elements of E are 2-element subsets of V. To avoid notational ambiguities, we shall always assume that $V \cap E = \emptyset$. The elements of V are the *vertices* (or *nodes*, or *points*) of the graph G, the elements of E are its *edges* (or *lines*)." (p.2)

Inspired by previous successful attempts to approach a location model as a graph (e.g., Kortuem et al., 2005; Ye et al., 2007a; Glassey, 2009), we have chosen to use graphs for the representation of our location model due to a number of reasons:

- Firstly, the simplicity and human-friendliness requirements identified in section 2.2, p.20, dictate that we should look for a visually simple representation which will be convenient to work with during the application design phase.
- Secondly, the uncertainty requirement dictates that we should look for a representation that can equally well manipulate both distinct (i.e. certain/complete) and vague (i.e. uncertain/incomplete) types of information.

- Thirdly, the heterogeneity of a pervasive environment dictates that our representation has to be able to represent an arbitrary resource.
- Fourthly, we need a data structure that is able to connect resources to each other, thereby forming the required frame for reasoning.
- Finally, our representation must have a means of connecting to the rest of the world, and graphs' abstraction to ontologies (that is, an ontology can be represented as a graph) fulfils this bit: in section 3.4, p.49, while discussing modelling with ontologies, we explain, in particular, that ontologies describing different bodies of knowledge can be combined and/or integrated with other domain ontologies.

Thus, each node of the spatial graph corresponds to an arbitrary resource and each edge stands for an arbitrary spatial relationship between two resources. Each edge may additionally be directed, i.e. connecting node A with node B but not vice versa (similar to a one-way road connection). Besides, there can be several edges, both directed and not, between the same two nodes. We can further label nodes and edges with individual properties, which would carry additional information describing them. For example, in Figure 3.3, each node is given a human-readable name of the resource it represents, and each edge is labelled with the distance between the two nodes it connects.

Altogether, the following changes are possible to the graph's structure:

1) a new graph component, either a node or an edge connecting any two nodes, appears;
2) an existing graph component, either a node or an edge connecting any two nodes, disappears;
3) the value of a property assigned to a graph component, either a node or an edge, changes.

By observing how the spatial graph of a given system changes over time, we are able to track the spatial behaviour of the resources.

3.1.3. Node availability

The advantage of assigning individual properties to the components of a graph at run-time lets the nomadic nature of the behaviour of resources in a pervasive environment be reflected in a spatial graph: we can model and analyse the spatial availability of each node in the current graph (and hence, of the corresponding resource in the environment). In the remainder of this sub-section we define and discuss about a node's spatial availability.

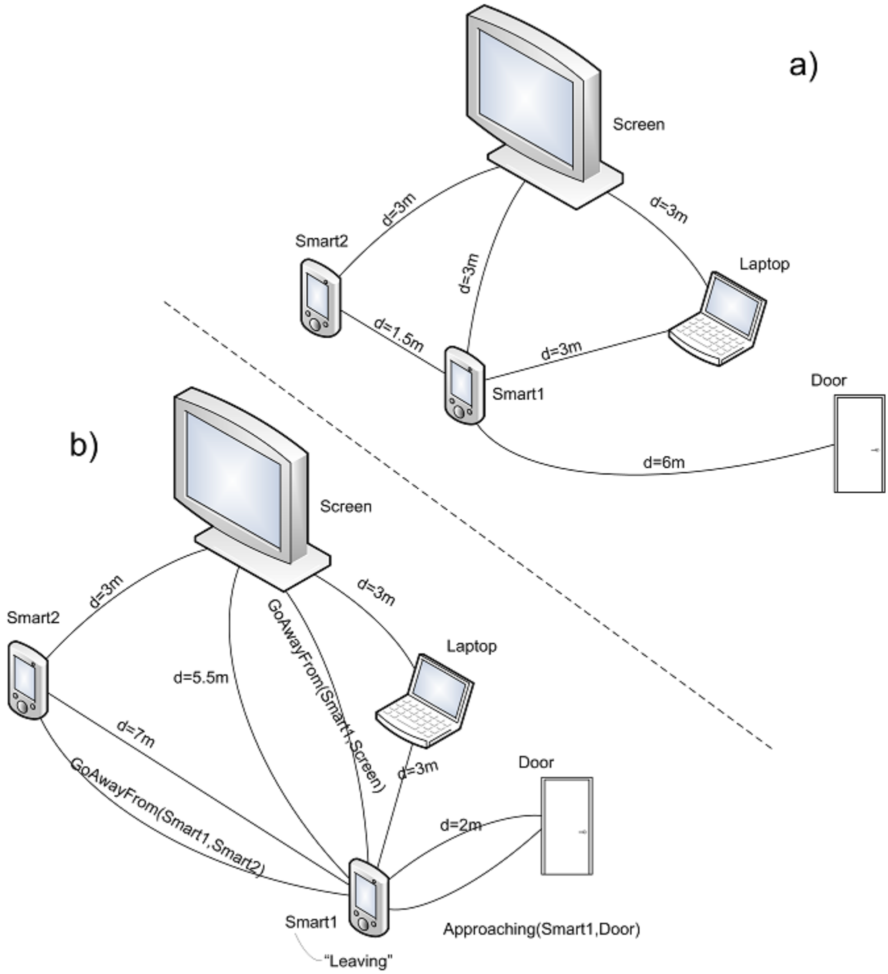


Figure 3.4. a) spatial graph of a system at time t_1; b) spatial graph of the same system at time t_3(> t_1).

Figure 3.4 depicts two layouts of the initial spatial graph from Figure 3.3 with an additional node representing the entrance door to the room. Figure 3.4a) shows the graph at an initial time moment t_1. At some later time moment t_2(>t_1), smartphone *Smart1* has moved further from the screen and closer to the entrance door. In the situation at yet a later moment t_3(>t_2), shown in Figure 3.4b), *Smart1* has moved yet further away in the same direction, which is reflected in the values of the distance relationships between node *Smart1* and the other nodes.

A quick analysis of the changes in the graph layout at times t_1, t_2, and t_3 identifies a dependency *d(Smart1,Door)* of the distance between smartphone *Smart1* and the entrance door over time. Figure 3.5 visualises a linear approximation of this dependency from time t_1 to time t_3 (the solid line), from which we conclude that *Smart1* is leaving the area. The conclusion yields a spatial relation between *Smart1* and the door – *(Smart1, Approaching, Door);* it also assigns *Smart1*'s internal property with a new state *"Leaving"* (see Figure 3.4b)). The other changing distances similarly produce spatial relations *(Smart1, GoingAwayFrom, Screen)* and *(Smart1, GoingAwayFrom, Smart2).* Extrapolating the plot for *d(Smart1,Door)*, we identify the expected behaviour of *Smart1*: it is expected to leave the interaction range (i.e. the conference room) at time t_4. Since the smartphone is currently carrying the zooming functionality of the image viewer's user interface, the system has to prepare for *Smart1*'s absence and determine where the corresponding part of the user interface should migrate when *Smart1* is no longer in the room.

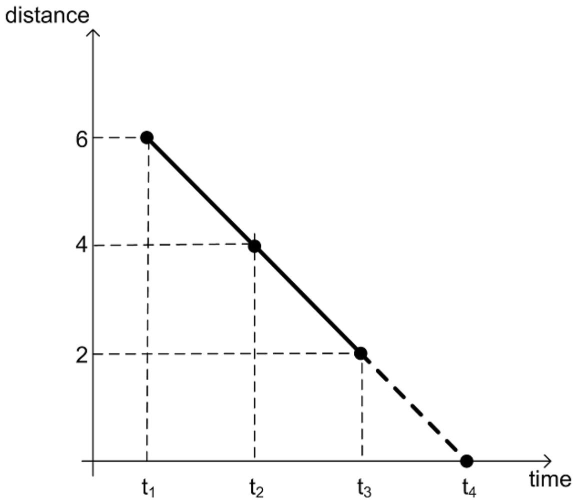


Figure 3.5. The dependency of the distance between resource *Smart1* and *Door* from time t_1 through to time t_3 is observed. It is extrapolated to a future moment t_4 – the time when the resource is expected to leave.

An issue with using discrete time intervals for representing continuous behaviour is that they need to be small enough so that no context is lost in between. In the spatial graph's terms, if no updates have been made to the graph components during time period [t_A; t_B], then the graph hasn't lost any context during this period. Following these terms, intervals [t_1; t_2] and [t_2; t_3] in Figure 3.5 are such time periods: we know the distances only at times t_1, t_2 and t_3, and the continuity of their changes is, as we noted, an approximation. The described attitude has limitations on the validity of its reasoning in time-critical situations but it is sufficient for our aim to identify a need for a user interface redistribution event.

Availability is an important aspect of consideration in a pervasive environment with its nomadic resources. The system should be able to identify and take into account the devices that are leaving, i.e. for which corresponding spatial relationships or properties exist. This way, only devices known to remain in the area at a given moment will be considered as potential successors, thereby causing a smaller interruption in the currently active user interface configuration.

Now, consider a situation when several devices are leaving simultaneously. If, for example, two devices, D_1 and D_2, have been identified as leaving, but D_1 has only started moving in the direction of the exit whereas D_2 is already close to the door, then it would be unnatural to treat them equally – D_1 will still stay available in the environment for quite some time, if decides to leave at all. Thinking this way, we can say that while D_1 and D_2 are both available, D_1 is more available in the spatial terms than D_2. Below we explain how we treat the above mentioned difference between nodes' spatial availabilities within the context of our definitions.

Fuzzy availability

The proposed extension to reasoning about availability is based on the concept of a fuzzy set, originally defined by Zadeh (1965):

> "...Let X be a space of points (objects), with a generic element of X denoted by x. Thus, $X=\{x\}$.
>
> A *fuzzy set (class) A* in X is characterized by a *membership (characteristic) function* $f_A(x)$ which associates with each point in X a real number in the interval [0,1], with the value of $f_A(x)$ at x representing the "grade of membership" of x in A. Thus, the nearer the value of $f_A(x)$ to unity, the higher the grade of membership of x in A. When A is a set in the ordinary sense of the term, its membership function can take on only two values 0 and 1, with $f_A(x)=1$ or 0 according as x does or does not belong to A..." (p.339)

We use fuzzy sets to help us treat a resource's behaviour after it has been identified as a leaving resource. Following the membership function's definition, we distinguish three zones of spatial availability (presence) of a resource in the environment: fully available (fully present), fully unavailable (fully absent), and partially available (i.e. has a fuzzy membership). A resource is considered to be fully available at all times prior to the identification of the leaving relationships. As long as these relationships and states exist while the resource is still available, we speak of its partial availability in the environment. Thus, availability is initially defined by the extrapolation we introduced earlier, so that before time t_3 the resource is fully present (times before t_3 in Figure 3.5; $d(t_3) \rightarrow a=1$), after time t_4 the resource is expected to be fully absent ($d(t_y)=0$ for all $t_y>t_4 \rightarrow a=0$), and between t_3 and t_4 its availability becomes fuzzy, with values from 1 at time t_3 down to the expected 0 at time t_4 ($0<a<1$ for $t_3<t<t_4$). Using these terms, we are able to say that a resource that is also leaving the area is a less reliable candidate, thus giving our preference to more stable ones. As noted, this reasoning applies only as long as the leaving relationships exist, and it comes back to $a=1$ otherwise (e.g., the resource stopped moving or is no longer approaching the exit).

In general, the concept of fuzziness can be applied to any context. If we, for example, consider the operational time of a mobile device's battery, its low charge status is a risk of this device's going offline, and upon detecting the critical level the system is forced to act in order to prevent information losses. We apply similar reasoning to the spatial availability. We slightly modify the initial approach to the fuzzy membership behaviour and introduce a minimum threshold for the availability's fuzzy membership, below which the lowness of the resource's availability becomes a threat, and therefore an action must be taken. Similarly, there is a maximum threshold, so that the resources with availabilities in some neighbourhood of unity are treated equally available. The logic behind the minimum threshold is related to our discussion about the applicability of discrete time intervals to representing continuous behaviour, so that we may not be able to catch the moment of absence, and the overall delay would turn out to be longer. The view on the maximum threshold is supported by usually weak differences between availabilities at their initial stages. Thus, both thresholds are context-dependent. Their determination is based on the application purposes and the user interface functionality the resource in question is carrying. Therefore, it is necessary to identify how important the corresponding resource is to the system; in other words, to know the price of losing the corresponding node of the spatial graph. The next section highlights on this bit.

3.1.4. Node importance

When a resource is leaving, we need to know how critical its loss is to the system. In this regard, a node's importance to the graph is a measure of the corresponding resource's irreplaceability in the system at this moment.

The concept of node importance is based on two things: application-based factors and resource-based factors. The first category reflects the varying importance of different functional components on the continuity and completeness of the application functionality. For instance, the ability to view images would most likely matter more than the ability to zoom a part of the currently displayed image. The second, resource-based, category takes into account the presence of active and available resources with similar capabilities. For instance, if there are several devices with a touch screen, then losing one that is currently used to perform the touch screen operations is not severe to the system, for its functionality can be given to any of the similar devices.

The process of importance determination is composed of a design-time and a run-time phase.

During the design-time phase,

I.A) Firstly, each part of an application's user interface is assigned a rank of its importance to the application in question (cfr. viewing images vs. zooming images):

$$App = \{UI_1, \ldots, UI_N\}$$
$$I_K = Importance(UI_K),\ I_{APP} = Sum(I_K)$$

This is done at the application's creation time by the application's designers.

I.B) Secondly, UI_K's means of input/output (e.g., a touch screen, a microphone) defines each device's (and hence node's) capability to hold this UI_K, which can also be known in advance:

$$N_A(UI_B) = \delta_{AB}, \text{ where } \delta_{AB} = \{1,0\}.$$

During the run-time phase, i.e. when a node (N_J) is leaving,

II.A) Thirdly, we determine which parts UI_C's of the complete interface currently belong to N_J:

$$UI(N_J) = \{UI_C, \text{ for all such } C \text{ that } UI_C \text{ is-on } N_J\}$$

II.B) Finally, we calculate the cost of losing node N_J.
To do so, we first find out how many non-leaving devices can hold each UI_C from the $UI(N_J)$ collection we determined in step II.A):

$$S_C = Sum_R\ (\delta_{RC}, \text{ for all such } R \text{ that } N_R \text{ is not leaving})$$

If at least one S_C equals 0, then the cost of losing N_J equals 1:

$$L(N_J) = 1,$$

for the application's functionality will lose its completeness. Otherwise, the cost of losing each particular UI_C is

$$L_C = I_C / S_C,\ S_C \neq 0$$

And the cost of losing N_J, or N_J's importance in the graph at the moment, is then

$$L(N_J) = Sum(L_C) / I_{APP}$$

This approach to calculating importance takes into account both the number of user interface components held by the leaving resource and the number of remaining resources that are able to take over at least one of those components. Besides, we also consider situations when there are no resources to send the leaving user interface component to ($S_C=0$). In this case, the importance reaches its maximum (i.e. worth the entire application) value because the application functionality's completeness will be affected.

The obtained expression for a node's importance takes, like availability, values in the interval [0, 1]. The intention behind the computations of availability and importance is to determine a scale grading that should be applied to the leaving resource's spatial changes. While availability mainly depends on the behaviour of the resource this node represents, importance analyses the situation in the rest of the environment. And a node's importance corresponds to this node's minimum availability, upon reaching which the system acts. Therefore, the more important the resource that gets out of scope is to the system, the more available its node has to be by the moment the disappearance is noticed (Figure 3.6: t_s is closer to t_3) and the sooner the corresponding resource's functionality is sent to the remaining resources (Figure 3.6: t_d is sooner).

Please note that we do not focus on the acting itself, e.g., on such aspects as interface migration, automatic generation, run-time adaptation, rendition, etc. Instead, we rather determine in which spatial situation the system should do so and whether the available resources are spatially reliable.

To illustrate how the calculations we have introduced affect the planned activities, we ran a simulated analysis in which we considered two configurations of the image viewer application and two different settings of the spatial graph.

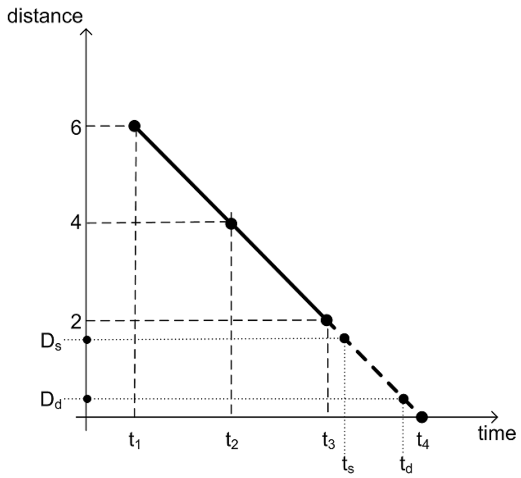


Figure 3.6. Higher threshold values mean that a more important leaving resource should stay more available when actions are taken.

3.2. Graph manipulation

The analysis is based on a collaborative activity of working with an image viewer application whose user interface is shared among several resources.

3.2.1. Importance

Resource *Smart1* holds the zooming functionality in the first distribution and the switching functionality in the second.

I.A): The corresponding importance values of the application user interface components are: *viewing – 3, switching – 2, zooming – 1, and typing – 2,* (the pie chart in Figure 3.7 illustrates their proportions within the application's complete interface).

I.B): Each device's capability to hold certain user interface functionality is also provided:

	Viewing	Switching	Zooming	Typing
Smart1	1	1	1	1
Smart2	1	1	1	1
Laptop	1	0(*)	1	1
Screen	1	0	0	0

(*)Switching has been realised for a touch-screen, so the laptop cannot do that, i.e. $\delta(Laptop, Switching)=0$.

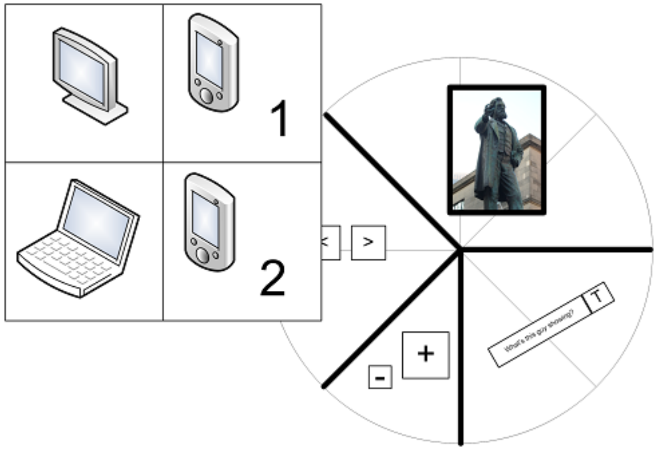


Figure 3.7. An application's user interface components have different weights, according to the functionality they provide.

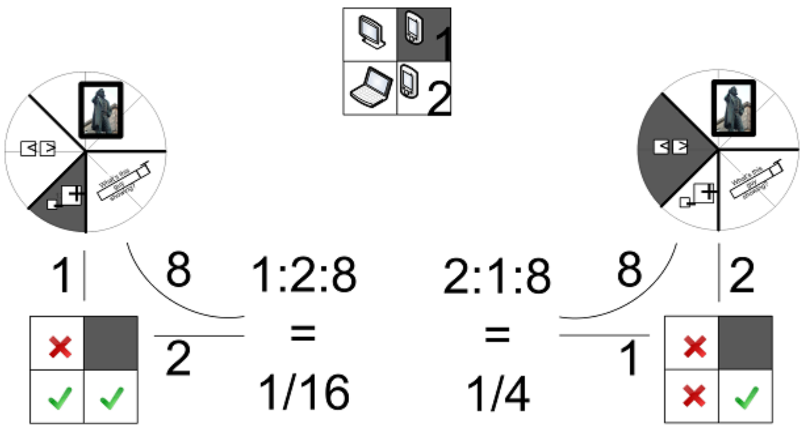


Figure 3.8. The importance of a leaving node considers both the weight of the user interface it is carrying currently and the number of nodes capable of receiving the leaving user interface.

The behaviour of *Smart1* in both situations is the same: it has been identified as a leaving node at time t_3. The interval $[t_3; t_4]$ is thus the time interval during which *Smart1* is partially available, with the availability value expected to decrease from 1 at time t_3 to 0 at time t_4.

Using the data provided in steps I.A and I.B, we compute that *Smart1*'s importance in the case of switching is *L(Smart1)=1/4*, and it is *L(Smart1)=1/16* in the case of zooming, respectively (see Figure 3.8).

As discussed, the obtained importance values correspond to the minimum availability, upon detecting which the system acts. That is, when *d(Smart1,Door)* reaches the corresponding D_d (see Figure 3.6, p.40).

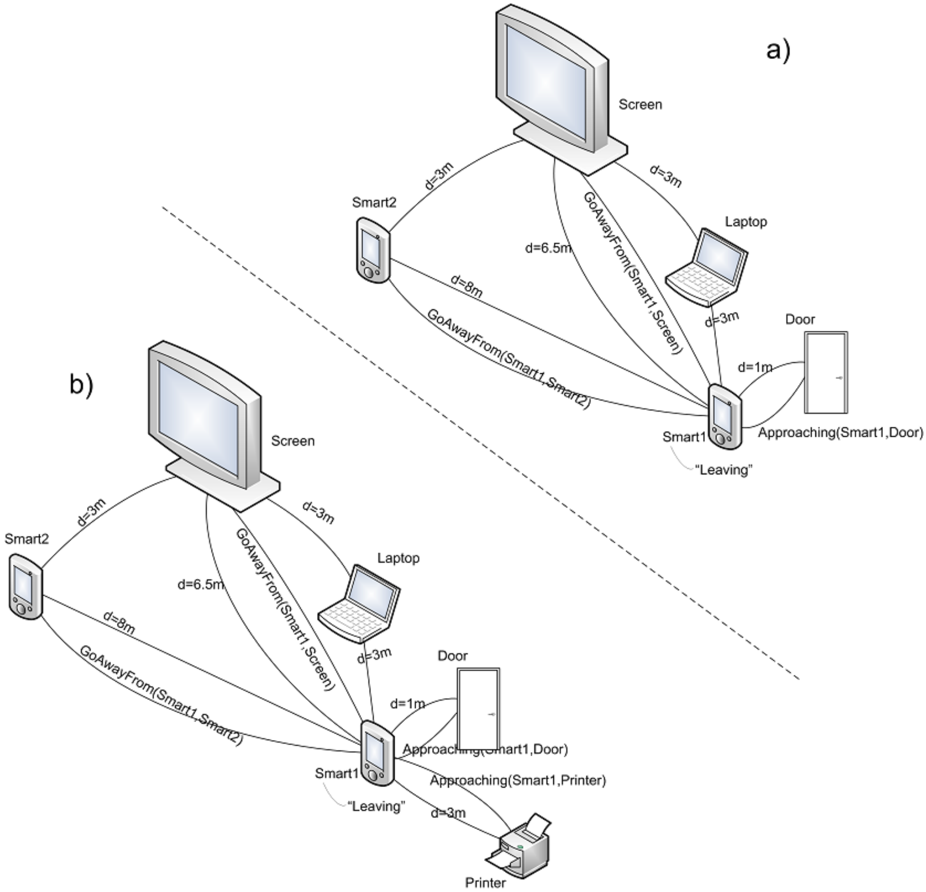


Figure 3.9. The activated printer has yielded an "approaching" relation for *Smart1* in the second case, acting as a holder and thereby increasing *Smart1*'s availability membership.

3.2.2. Availability

Now consider the behaviour of *Smart1* after time t_3 in two different layouts of the spatial graph: with and without an active printer in the room (see Figure 3.9). Its availability at time t_{3a} (see Figure 3.10) is 0.5 in the first case (situation in Figure 3.9a)), since no changes have happened to the pool of relations and the behaviour therefore has developed as expected. In the case shown in Figure

3.9b), the active printer yields an "approaching" relation when the distance between *Smart1* and *Printer*, *d(Smart1, Printer)*, decreases in a similar manner to *Smart1*'s approaching the door. The availability at time t_{3a} thus increases to 0.75: the new "approaching" spatial relation between the leaving resource and another resource within the area acts as a holder.

3.2.3. Discussion

A potential issue with the approach to determining availability (hence the thresholds) based on the extrapolated distance is its simplicity, which is not necessarily the most accurate approximation. However, a straight-line approximation rather overestimates than underestimates the time required to cross the distance. It does not take into account such factors as obstacles or pauses along the route, so that in reality a straight line simply represents the best case. But in the case of a longer time required, the proposed estimation of the critical availability remains applicable: it is affected by run-time spatial relationships, and its performance may only increase by filtering out outdated and considering newly added relationships.

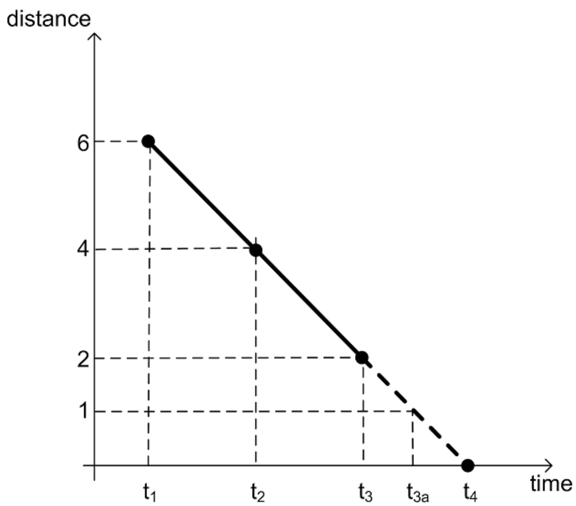


Figure 3.10. At time t_{3a}, the distance *d(Smart1, Door)* is expected to decrease to 1.

Every action in the system takes time. In this regard, the minimum threshold, beyond which the values are rounded up, depends, to a considerable extent, on the processing capabilities. If in Figure 3.10 a system needs at least $(t_4\text{-}t_3)/10$ time units to complete the action, then the determined earlier importance of

1/16 of losing the smartphone that holds zooming should be rounded up to the allowed minimal threshold of 1/10. Similarly, the maximum threshold is as much application-dependent as it is spatially so. For example, in the case of applications for people with dementia, the allowance of leaving a route should be taken care of much stricter, i.e. warnings shown earlier, if not immediately.

We have described and illustrated how to model spatial connections between resources in a pervasive environment and how to identify and handle situations of interest. We now need a view of the spatial world that not only would allow the resources to sense the presence of other resources around them, but also see. In other words, we need to extend the model to let the resources in a pervasive environment additionally know about each other's relative arrangements. The next section highlights on this bit.

3.3. Placing resources

Such factors as the relative position, the orientation, or the direction of resources' locomotion in the environment also have an impact on the way these resources interact with their surroundings. Knowing about the spatial arrangements *around* a resource is therefore an important step in getting a finer insight into the spatial topology of the environment and in providing better communication. This section presents and details the directional aspects of interaction between resources.

3.3.1. Ego- and allo-centric world view

The initial question "Where is entity N?" that we posed in section 2.1.1, p.13, may be answered, in particular, in the two following ways: "N is to your left" and "N is between A and B". Both answers point at the same object, but each uses a different frame of reference. In the first case the reference is the user themselves, and in the second case, the user refers to other objects. The former approach is referred to as the egocentric frame of reference (self-referenced), the latter is called allocentric (object-referenced). Figure 3.11 shows an example of each approach applied to the collection of resources we addressed in section 3.1.1.

Discussions and investigations of the advantages, effectiveness, applicability, etc. of both approaches to various scenarios and in different environments still take place (e.g., McNamara et al., 2003; Burgess, 2006). In particular, whereas familiar environments tend to be represented in memory and addressed in the allocentric way (e.g., Ruggiero and Iachini, 2006), an unknown area is mainly viewed egocentrically, employing the more natural way of communication between an individual and environment (Iachini and Ruggiero, 2006). We, too, advocate the egocentric approach as the more intuitive one with respect to referring to other resources in the environment during interaction.

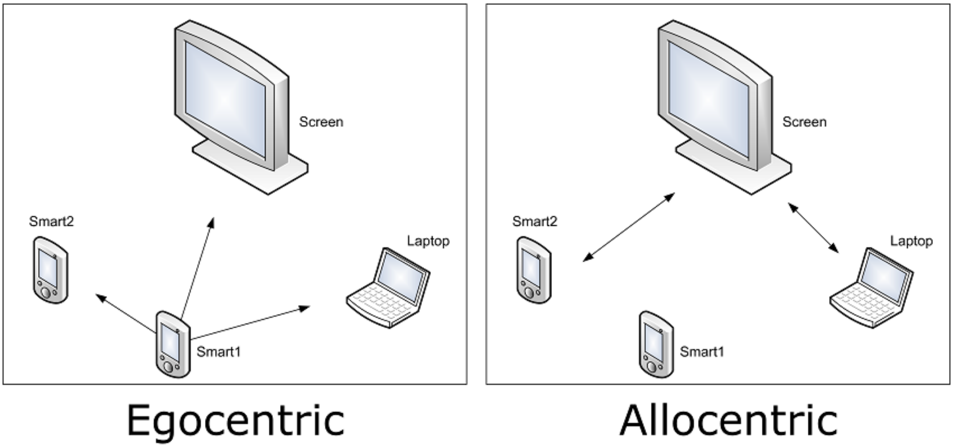


Figure 3.11. In the egocentric approach, the reference is the user themselves; in the allocentric one, an object is identified by referencing to other objects in the environment.

We describe in the following sections an ego-centric world view of the environment around a resource and establish relative positional relationships that apply between resources. An important component of the model is that it takes into account uncertain and imprecise spatial knowledge, preserving the simplicity and understandability of the model. This type of knowledge allows spatial arrangements between interacting resources be considered in a more natural way, thereby improving the overall spatial awareness of resources in a pervasive environment.

3.3.2. The Ambient Compass

The first two requirements for pervasive location modelling identified in section 2.2 suggest that a suitable model should not be overloaded with concepts and should be easy to understand. Following these, we propose a classification that aims at giving an application the possibility to speak a language similar to that of humans when they refer to spatial arrangements between objects in space.

The model that we called *the Ambient Compass* defines a set of basic concepts natural to spatial structures. These include a 2D relative position and orientation, followed by a division of the space around a resource into eight zones as depicted in Figure 3.12. This division resembles the way we generally refer to the parts of the world: north, east, south, west, north-west, north-east, south-east, and south-west. The four cardinal zones – left, right, front, and behind – yield a corresponding cardinal spatial relationship – "hasOnLeft", "hasOnRight", "hasInFront", and "hasBehind" – between the compass owner and another resource with which the relationship is being established. The four other zones yield a pair of the corresponding cardinal relationships (e.g., "hasOnLeft" as well as "hasInFront").

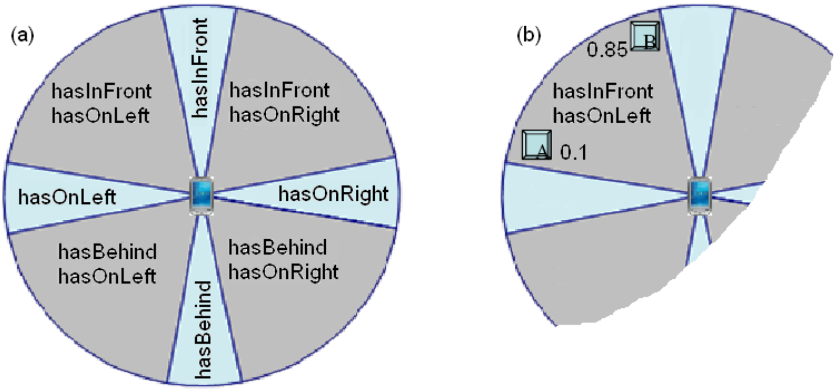


Figure 3.12. (a) The Ambient Compass divides the space around a resource into eight zones; (b) resources belonging to the same zone of the compass are distinguished by means of assigning each of them a degree of membership to this zone.

In the basic design, the boundaries of each of the main four zones are symmetric with respect to the left-right and front-below directions. The slope angles depend on the characteristics of a resource in question; they are assigned individually for each category of interaction resources based on their communication specifics. Additionally, each relationship of type "has" is accompanied by an inverse relationship of type "is", so that if we establish relation *(Resource1, "hasOnLeft", Resource2)* then we also identify its inverse relation *(Resource2, "isOnLeftOf", Resource1)*. This way, we cover both situations when we need to find out about a given resource’s arrangements with respect to others (the “is” relations) and about these of the other resources with respect to the given resource (the “has” relations). A similar computational model – also an ego centric frame of reference with distinguishing eight viewing directions – was proposed by Baus et al. (2001). It was applied in path visualisation tasks to help identify the presence of objects in the corresponding directions along a route in order to describe route instructions to a pedestrian.

In general, there are two vertical cardinal directions, "below" and "above”, that should also be considered. In this regard, we would like to consider two thoughts: firstly, Franklin and Tversky (1990) showed that people’s perception of objects in the "above/below" directions proved to be greatly asymmetric as opposed to, for example, the "left/right" pair. Secondly, a planar representation already suffices a wide range of popular categories of location-aware applications (e.g., mobile tourist guides are usually based on 2D positioning and maps (Kenteris et al., 2011), with the 3D representation used as an optional extension of the graphical representation), so that supporting a fully functional 3D scheme of relationships becomes unnecessarily computationally expensive and redundant (Glassey, 2009). Besides, in order to represent the third dimension for required scenarios, it is possible to consider a reduced symbolic support of elevation (e.g., Dürr and Rothermel, 2003) and apply a transformation between coordinate reference systems (e.g., Jiang and Steenkiste, 2002). Thus, within the context of this work we focus on 2D spatial models and relationships.

The presented Ambient Compass model acts as a basis, on which we build two extensions to make the model capable of handling relevant uncertain and imprecise knowledge about the spatial world.

3.3.3. Considering uncertain and incomplete knowledge

Circular extension – adding fuzziness

The first, circular extension deals with the ambiguity that appears when an addressed resource happens to be in a mixed zone of a referred compass. In Figure 3.12(b), devices *A* and *B* both belong to the "hasInFront-hasOnLeft" area of the compass, though their relative positions with respect to this compass' owner are different. Therefore treating *A* and *B* as spatially equal would be incorrect. A solution to this ambiguity lies in introducing fuzzy memberships of the resources to the corresponding zone (for fuzzy set and membership's definitions please refer to section 3.1.3, p.36). Devices *A* and *B*, as shown in Figure 3.12(b), are assigned different values (0.1 and 0.85, respectively), which grade their degree of being in front of the referred resource (fuzzy memberships are linked with the "hasInFront" direction). The two membership values of the corresponding complement relationships of a mixed zone sum to unity; that is, if a resource is at ¾ of "hasInFront", its left membership "hasOnLeft" is ¼. The same logic applies to the cardinal zones, in which a resource's fuzzy membership is computed clockwise within the zone.

The purpose of the proposed extension is to assign weights to the relationships in the spatial model. While preserving the original simplicity of relationships and keeping the easiness of deriving them, we extend the reasoning pool to reflect a more truthful actual relationship between a pair of resources.

Radial extension – bearing distances

The radial extension results primarily from the way humans perceive distances. The concept of closeness of one object to another varies depending on what factors one considers to be important in a given situation. The proposed extension is built on two components and combines two concepts of approaching spatial proximity.

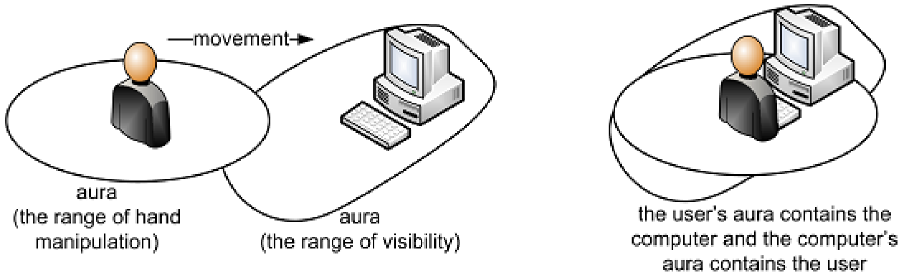


Figure 3.13. Interaction areas of user hand-manipulation and for range of computer's visibility, adapted from Satoh (2007).

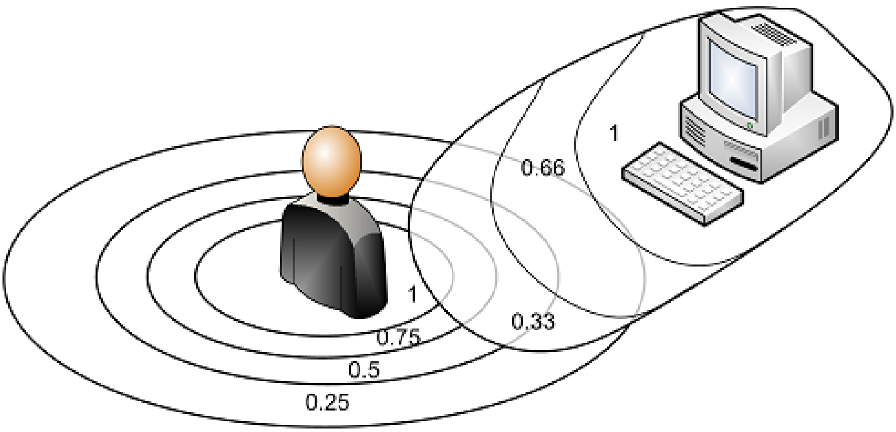


Figure 3.14. Proximity illustrated as a set of neighbourhoods, adapted from Geusgen (2002).

The first component, outlined in Figure 3.13, is based on Satoh's (2007) *aura* concept, originally specified as a virtual scope around a resource where interaction with this resource is possible. It is an application- and resource-specific component, so that its shape and size varies. For example, the interaction area of a touch table is within a hand's distance from the table border, the interaction area of a large screen in a room is the part of the room to which the screen is showing, etc.

The second component extends the concept of interaction area and introduces a "nearby" spatial relationship that defines a degree of closeness of other resources in the referred resource's interaction area (see Figure 3.14). Following the fuzzy membership concept, we determine how much a resource is "nearby" by assigning a real number in the range [0; 1].

Considering distances in such a way can provide solutions in situations where no perfect match can be found but still a positive response can be obtained (Guesgen, 2002).

3.3.4. Discussion

We have introduced and discussed how resources in a pervasive environment can be spatially connected and allocated into a spatial computational and reasoning model, and how situations of interest can be identified and handled in that model. We have also proposed a way to arrange and handle incomplete and uncertain knowledge. With the model at hand, we now need a tool that would allow us to operate the model in the context of a pervasive environment; that is, to manipulate the obtained data digitally, making the spatial knowledge part of a processing framework. The next section highlights on the techniques and tools used to operate a pervasive environment and how the spatial model can become part of it.

3.4. Modelling with ontologies

Ontology is a form of knowledge representation that organises information in a structured way, making it, in particular, understandable to machines. Gruber (1995) defines an ontology as follows:

> "An ontology is an explicit specification of a conceptualization" (p.908),

where a conceptualization is, in turn, defined as

> "...the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them." (p.908)

Ontologies have been applied in abundance to organise and categorise the available information within a diversity of application domains (e.g., ambient intelligence (Preuveneers et al., 2004), bioinformatics[6], linguistics[7], pervasive

[6] http://www.geneontology.org

computing[8], people's social profiles and connections[9], web-site categorisation (Labrou and Finin, 1999)). There exists extensive tool and framework support for working with and manipulating ontologies: ontology editors (e.g., Protégé[10]), reasoning engines (e.g., Pellet[11], RacerPro[12]), programming frameworks (e.g., Jena[13]), etc. Besides, studies about ontologies have emerged into a separate branch of research activities in the field of knowledge engineering.

Rephrasing the above Gruber's definition, we say that a typical ontology consists of (groups of) concepts and relationships that connect these (groups of) concepts. A concept can be anything that is of interest in the environment, and a relationship makes a statement about a (group of) concept(s) with respect to other (groups of) concepts. Following these terms, the spatial graph and the relationships infrastructure of the graph's edges and the concepts of the Ambient Compass can also be all together represented as an ontology.

In the following sub-sections we discuss how ontologies can be used to describe a pervasive environment operated, respectively, by an ontology-based framework, and then introduce and explain the details of integrating our spatial model into this ontology and framework.

3.4.1. Higher- and lower-level ontologies

Ontologies are said to be the most suitable means of representing context for pervasive applications (Gu et al., 2004).

One approach to modelling the knowledge about a heterogeneous and dynamic environment in an ontology is to employ two levels. At the lower level there is a domain ontology, which describes concepts specific within a domain. At the upper level there is an upper (top-level) ontology, which describes high-level domain-independent generic concepts. An example of an upper ontology is the Suggested Upper Merged Ontology (SUMO)[14], with its *Agent* or *Language* concepts, among many others, as examples of generic concepts applicable to a variety of domains. Recently, Ye et al. (2011) have proposed to take into account yet another level of abstraction and described what they called a top-level ontology. It divides any ontology's root concept, *Thing*, into the *Concept*, *Con-*

[7] http://www.linguistics-ontology.org

[8] http://ontonym.org

[9] http://www.foaf-project.org

[10] http://protege.stanford.edu

[11] http://clarkparsia.com/pellet/

[12] http://www.racer-systems.com/products/racerpro/

[13] http://jena.sourcefourge.net

[14] http://www.ontologyportal.org

text, and *Activity* notions common to all information spaces within a smart environment. The authors contend that the top-level ontology model, while beneficial, should not be seen as a replacement, but rather a complementary extension to the other two levels, whose aim is to reduce the required effort in engineering domain knowledge.

3.4.2. Upper environment ontology for run-time support

Different from most ontologies that were developed to support the creation and modelling of context-aware systems, the focus of Vanderhulst's (2010, chapter 2) semantic environment model and its upper ontology is to support and manage the functionality and adaptation of pervasive applications at run-time. The proposed upper environment ontology describes concepts common to pervasive applications, such as *Resource*, *User*, *Device*, *Service*, *Task*, *User Interface (UI)*, etc. (see Figure 3.15 for the complete layout). The environment ontology is the core part of the environment model, and domain ontologies describing concepts of an application domain are attached at run-time, according to application needs.

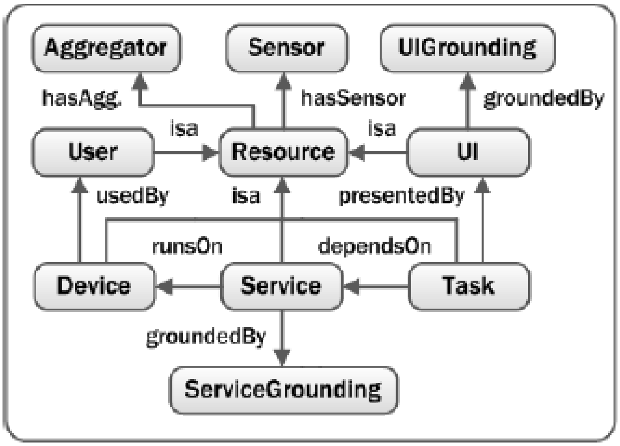


Figure 3.15. An upper environment ontology used by Vanderhulst (2010), reproduced from Vanderhulst (2010).

The environment ontology is built in the Web Ontology Language (OWL)[15], which is a standardised mark-up knowledge representation language designed to work with ontologies and based on the Resource Description Framework (RDF)[16]

[15] http://www.w3.org/2004/OWL/

[16] http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

specification in the domain of Semantic Web. In OWL, objects, concepts and other entities are defined as *classes*; instances of one or more classes are called *individuals*; the relationships that hold among them are held in *object properties*, and *datatype properties* define relations between individuals and RDF literals and XML Schema datatypes, such as strings, numerical values, etc. Besides, OWL distinguishes between three levels of expressiveness: OWL Lite, OWL DL, and OWL Full. The environment ontology adheres to the OWL DL version, as to one that has the maximum expressiveness and at the same time guarantees that all entailments can be computed in finite time. Besides, in the frames of OWL DL we are able to establish class equivalence between similar but independent concepts, which is important in the case of multiple ontologies.

3.4.3. ReWiRe

The previous section's environment model, described using the corresponding upper environment ontology, is managed by ReWiRe, a middleware infrastructure for managing pervasive applications, as well proposed by Vanderhulst (2010, chapter 5). It relies on the OSGi framework[17], which is a Java-based platform for modular application deployment, and uses the Apache Felix[18] implementation to manage the life cycle of its OSGi bundles.

The architecture of ReWiRe comprises a host, which is a server side that stores models and serves as an execution environment, and clients, an instance of which is installed on each device that participates in the environment. A collection of default OSGi bundles (Environment, Behaviour, Jena …) form the framework's initial functionality, so that it is ready to discover and work with pervasive applications that are built ReWiRe-compatible and are installed on potential client devices.

In order to work with ReWiRe, a pervasive application must have a domain ontology, a corresponding service (or several services), and user interface components to observe and interact with the service in question. A domain ontology extends the concepts of the upper environment ontology and describes this application's domain in detail, as appropriate.

The ability of the environment model managed by ReWiRe to adapt to changes at run-time makes it a suitable framework for supporting our pervasive location model. We incorporate the spatial model as a spatial domain ontology aggregated with the environment ontology. Thus, we make spatial context a part of a processing middleware for managing pervasive applications based on the above environment model, incorporating it as a spatial service into this mid-

[17] http://www.osgi.org

[18] http://felix.apache.org

dleware. The next sections introduce a spatial domain ontology that incorporates the pervasive location computational model as described in sections 3.1–3.3 and describe its integration as a spatial service into ReWiRe and its user interface.

3.4.4. Spatial ontology

Each spatially meaningful resource in the environment is a SPATIALRESOURCE, which extends the RESOURCE concept of the upper environment ontology. Each SPATIALRESOURCE has a LOCATION and ORIENTATION. Both LOCATION and ORIENTATION are linked to a REFERENCEPOINT, which is a means to connect location engines (such as coordinate systems or symbolic location results used to locate a SPATIALRESOURCE), using approaches as explained, for example, in (Jiang and Steenkiste, 2002; Stevenson et al., 2010).

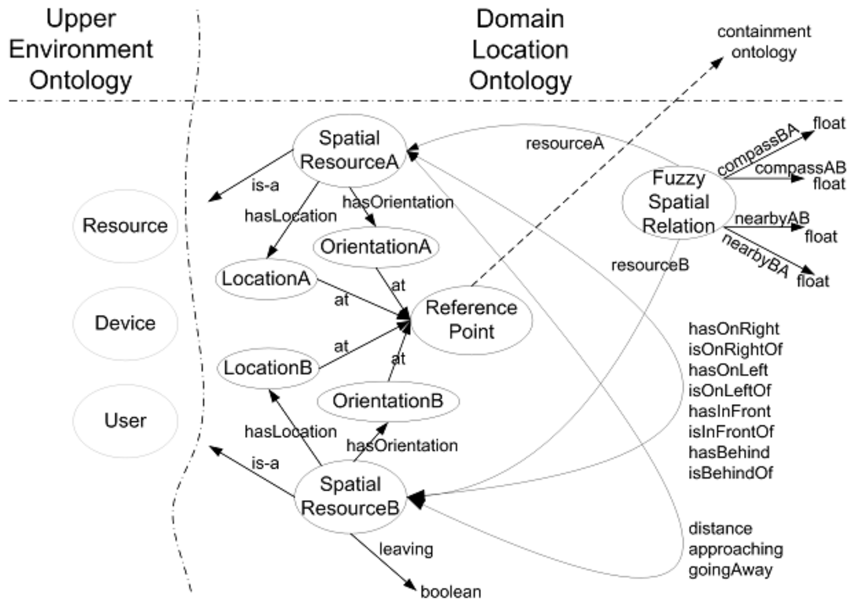


Figure 3.16. The spatial ontology is a domain ontology of the upper environment ontology.

When we know the locations of the resources in question, we establish the "distance" and the two movement relationships – "goingAway" and "approaching"; and when we learn the orientations, we compute the group of the Ambient Compass' relationships.

To deal with fuzziness in the Ambient Compass, we introduce the concept of a FUZZYSPATIALRELATION. This approach was inspired by the concept of a Fuzzy Relation that Gu et al. (2007) introduced in their edutainment ontology. Their Fuzzy

Relation has two object properties, each linking to the corresponding "universe of discourse" (p.592), and a datatype property for the value of the fuzzy membership. The self-evidence and computational simplicity of this representation fits to describe the established fuzziness in the spatial relations as well. Therefore we introduce the concept of a FuzzySpatialRelation into our spatial ontology. It has two object properties, one per spatial resource, and two pairs of fuzzy membership datatype properties, per compass and per nearby, respectively. Note that one instance of FuzzySpatialRelation is linked to two resources and stores the corresponding fuzzy memberships of one resource with respect to the other. The complete layout of the spatial ontology is depicted in Figure 3.16, and Figure 3.17 shows an extract of the spatial ontology created in Protégé, extending the upper environment ontology.

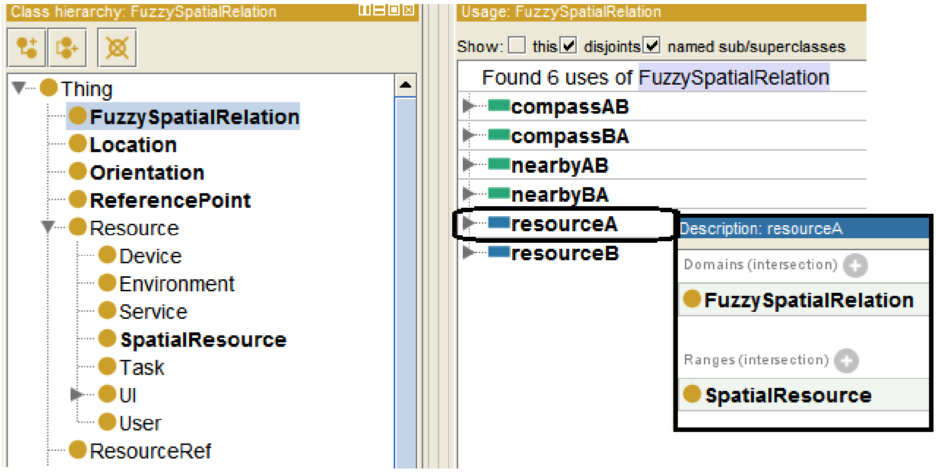


Figure 3.17. The spatial ontology created in Protégé, extending the upper environment ontology.

In the next section, we explain the integration of the spatial ontology model as a SpatialService into ReWiRe and the addition of an environment visualisation plug-in into the user interface to manipulate and observe movements of spatial resources.

3.4.5. Spatial service in ReWiRe

The spatial ontology, aggregated with ReWiRe's upper environment ontology, enriches the environment with the location context of resources. In accordance with the ReWiRe architecture, the support of location information is based on a combination of the domain spatial ontology and a corresponding OSGi bundle, which can be activated and deactivated at run-time (Figure 3.18B), and is deployed as a SpatialService.

When the spatial bundle is active, i.e. the ReWiRe client is configured to receive resources' location updates, the movement of the resources throughout the environment can be observed via a map-based plug-in that we added to the ReWiRe client's user interface (Figure 3.18C, Figure 3.18D). Using the provided query tool (Figure 3.18E), we can inspect spatial arrangements by using SPARQL queries (e.g., see Listing 3.1).

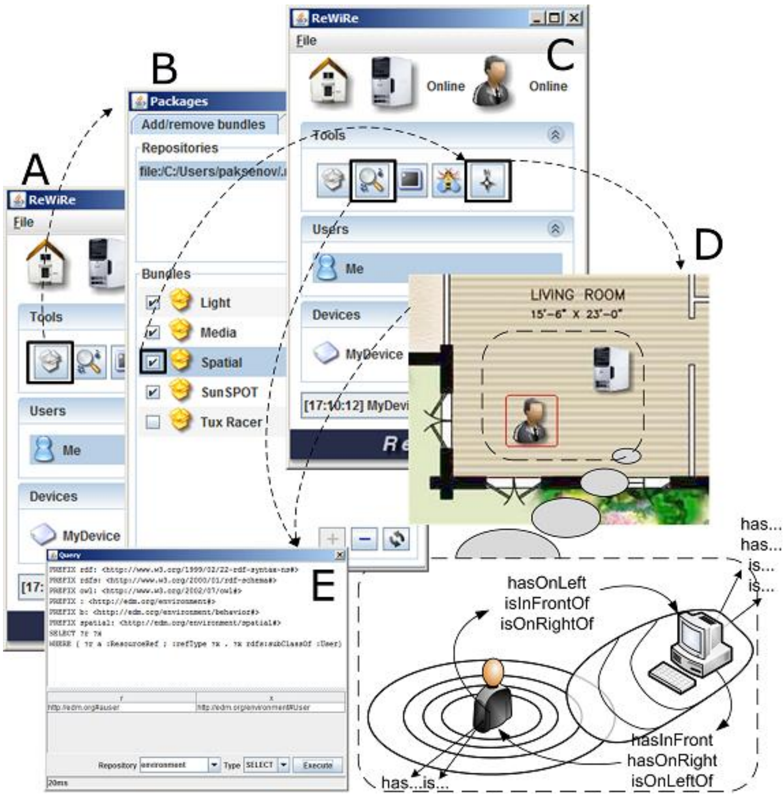


Figure 3.18. The support of location model in ReWiRe is based on a spatial bundle that can be activated and deactivated at run-time (B). The movements of resources can be observed in a map-like visualisation plug-in (D) invoked in the ReWiRe client's user interface (C). The querying tool (E) allows one to inspect spatial arrangements between resources by using SPARQL queries.

```
...
PREFIX e: <http://edm.org/environment#>
PREFIX s: <http://edm.org/environment/spatial#>

RESOLVE {
SELECT ?r WHERE { ?r a :ResourceRef . ?r :refType s:SpatialResource }
}
SELECT ?x WHERE { ?r1 s:ref <R1_URI> . ? r1 s:hasInFront ?x }
...
```

Listing 3.1. A SPARQL query asking for all resources that are (at least) in front of a given resource with the URI value “R1_URI”.

The movements’ visualisation plug-in has two modes: the simulation mode and the sensing mode. In the simulation mode, we can drag resources along the map and change their orientation in space. The sensing mode is the real location-aware mode, in which location update events in the model are triggered by actual sensor readings of the corresponding resources as they move. In both cases changes can be observed and their impact on the spatial arrangements monitored. The second requirement for a pervasive location model identified in section 2.2 calls for a human friendly model and interpretation. The proposed realisation allows developers, designers, and users of the resources connected with ReWiRe to use the same language during both application design and running phases, respectively. This additionally assures that all changes that happen while the application is already running can be interpreted in the way they are used at the design step.

3.4.6. Example usage: switching between screens

As an illustration of exploiting the spatial context, consider the Ambient Compass of a user’s smartphone in a room with two adjacent large screens, as it is shown in Figure 3.19.

The user is currently watching at screen A, so the smartphone’s Ambient Compass is oriented accordingly and the application is projecting the viewing part of the application’s user interface on the screen. Screen B is in the compass’ mixed zone and is therefore on stand-by (indicated by a dashed line in Figure 3.19a).

At a certain moment, the user starts rotating clockwise. As they are doing so, the smartphone’s "hasInFront" relationship with screen A changes its fuzzy membership value in accordance with the circular extension (see section 3.3.3). Screen B remains in the mixed zone, but the changes are also reflected in the fuzzy membership of the screen to the Ambient Compass’ mixed zone. When the rotation reaches an equal angle with both screens, they are determined to be in the front zone (Figure 3.19b). In this case the application wakes screen B up (the dashed line becomes solid in the figure) to send it the current view as well, so that the view is now shown on both in the front zone. As the turning continues, screen B centralises in the front zone, which is reflected in the value of this screen's fuzzy membership to the front zone, whereas screen A ends up in the mixed zone (Figure 3.19c) and is turned to stand-by.

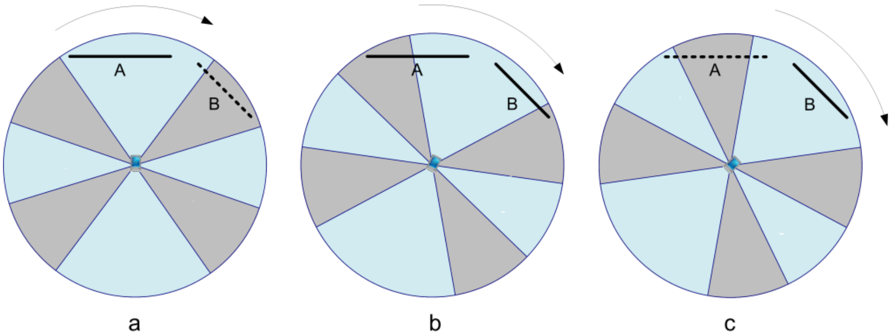


Figure 3.19. Observing the spatial changes caused by a smartphone rotating clockwise, the application changes the statuses of screens A and B from inactive (dashed line) to active (solid line), and vice versa, in response to the rotation. The change of the active screen is preceded by the state when the image is shown on both of them.

This example illustrates, in particular, how this sort of spatial awareness, i.e. fine-tuned simple spatial relationships, can be used to smooth the procedure of, for example, re-distributing a pervasive application's user interface.

3.5. Considering location data sources

We have assumed so far that resources are exactly at the determined locations. However, obtaining perfect location information is a very expensive challenge, rarely affordable. Moreover, the heterogeneous and frequently changing nature of a pervasive environment usually complicates this process even further. As a result, location information becomes imperfect, thereby impeding the easy adoption of the spatial model. It also brings along other issues to consider, such as when a resource's actual location may not belong to the calculated compass zone, may have a significantly different fuzzy membership value within that zone, or may simply be insufficient for calculating the spatial relationships at the desired level of granularity. This, in turn, causes incorrect assignments and so brings confusion and dissatisfaction. Therefore the spatial context must also take into account the differences that the determination of locations brings in.

3.5.1. The varying nature of location determination

In Figure 3.20 (top) an outdoor user is tracked by GPS and finds their way to the destination building successfully.

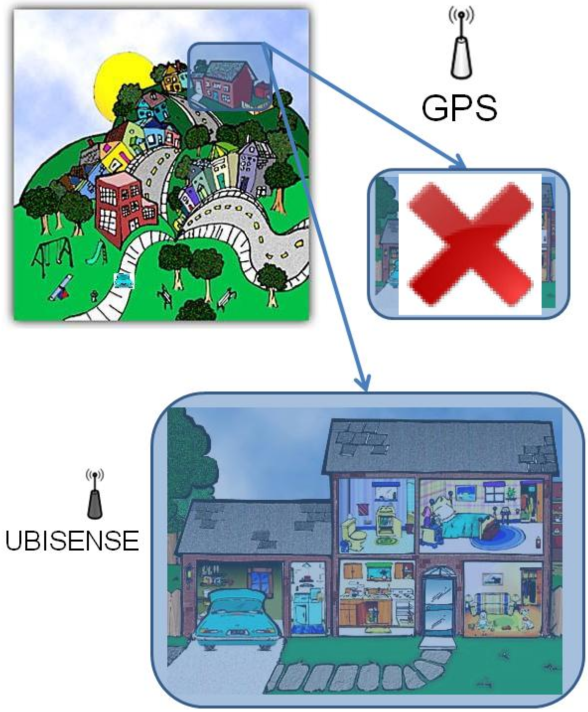


Figure 3.20. The outdoor area and the building are controlled by different location systems and different type of location data is sent to the application in each case. The creation of the figure was inspired by, and is based on, the images seen on the web website of the U.S. Environmental Protection Agency (http://www.epa.gov/kidshometour/) [accessed 11 Jan 2012].

In the house, the user decides to use a screen to see the photos just taken with the smartphone. Kept being located by GPS (top-right), the smartphone cannot get a location at all, for GPS normally does not work indoors[19], and the application cannot start the slideshow because the smartphone is detected to be outside of the screen's interaction area (i.e. its membership to the screen's "nearby" interaction area is 0). But if, for example, located by Ubisense, a precise real-time location tracking system installed in the building (bottom), the de-

[19] Even when it does – which is, as Kjærgaard et al. (2010) have shown, is sometimes possible – the returned location is less accurate and the behaviour is rather unstable.

termined solution would be correct since Ubisense is usually accurate enough (see section 2.3.1, p.21). Therefore not only is it important to know about the current location, but we also should stay aware of and be able to react to its limitations.

3.5.2. Impact of sensing on the location model

Ambient compass' layout

The Ambient Compass itself does not take into account the quality of location measurements. However, in this case it is not clear how the application should adapt its calculations when location determination is taken over by another provider and the layout of the Ambient Compass and the relationships should be reorganised. Without knowing the corresponding characteristics of the location system currently in use, such a transition between two states of the compass or interaction areas will likely lead to erroneous and unsuitable results. Awareness of this type of change lets the application adjust the layout of the compass and interaction areas and connect the interacting resources in an appropriate way. Figure 3.21 shows, schematically, an example of different layouts of the Ambient Compass as a result of location sensing capabilities, yielding a different set of spatial relationships calculated for the same arrangement of interacting resources in the environment.

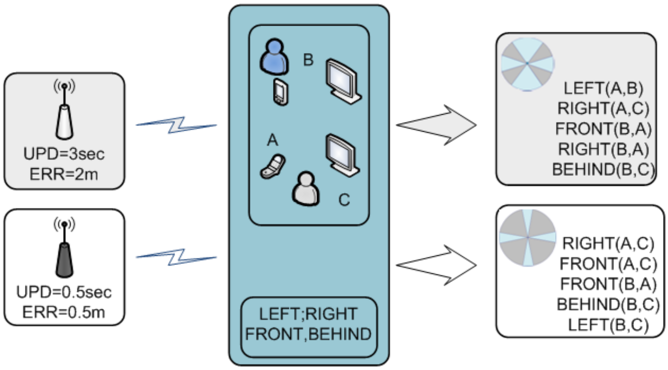


Figure 3.21. Different location systems have different characteristics. Based on these characteristics, the layout of the ambient compass is defined and spatial relationships are calculated as appropriate.

Insufficient data

We initially tested different arrangements of the same scenario from Figure 3.19, with the help of a room-scale Ubisense set up. A set up consists of a set of firmly fixed (ubi-)sensors, and the locations of (ubi-)tags are tracked throughout a certain area defined by the position and orientation of the sensors. We used a "Ubisense Research Package", comprising four sensors and ten tags, which we installed and configured in an area of approximately 3.5m by 3.5m. The system's ability to change the update rate, i.e. how often the system reports on a tag's location, lets us test different tracking conditions. Our preliminary verifications of the Ubisense installation revealed that the original scenario could not be completed because of inability to reflect rotation of a device of the size of a smartphone while at the same location. Therefore we altered the scenario to make it consider the same smartphone moving along three displays, as shown in Figure 3.22, with the same action of shifting the active screen as the smartphone moves from one to the next.

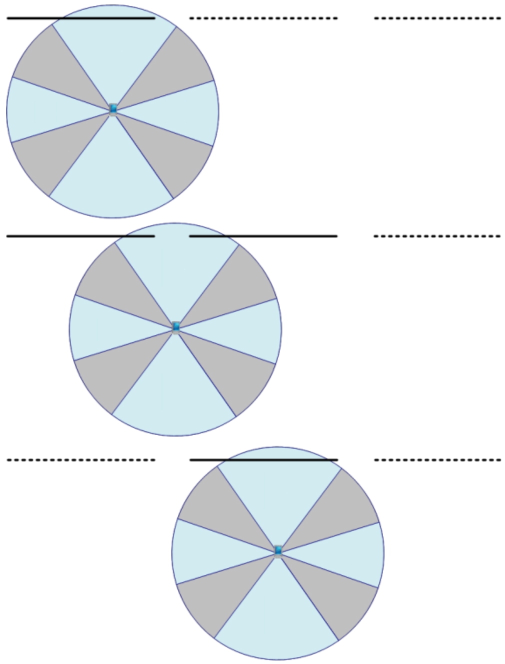


Figure 3.22. As the smartphone oriented towards the screens moves from left to right, the state of these screens changes accordingly.

3.6. Discussion

This chapter discussed how to model, represent and manipulate spatial arrangements between resources in a pervasive environment. We have integrated the proposed spatial model into an ontology-based framework for managing pervasive applications called ReWiRe. An important aspect of the model is its ability to addresses the dynamic and heterogeneous nature of such environments by taking into account uncertain information about spatial arrangements. The ability of ontologies to be combined, together with ReWiRe's feature to welcome various contexts, represented as ontologies, at run-time, makes location support an integral part of the managing framework, hence of pervasive applications. The model's other important advantage is its similarity to a developer's, designer's, and user's viewpoints altogether. This assures the correct understanding of the modelled information by each involved part, i.e. during each of a pervasive application's design, creation, and usage phases.

We have also experienced that the variability in pervasive location provision capabilities can (strongly) affect the location model and be the reason of inconsistencies. The changes that we were obliged to introduce in the verification scenario only strengthened our thoughts that incorporating the knowledge about localisation systems' fine details into the framework was necessary. In the next chapter we present and discuss our approach to considering any localisation system in a pervasive environment.

Chapter 4

Arrangements for pervasive location sensing modelling

Parts of this chapter have been published in the following conference proceedings and a journal article:

- Aksenov, P., Luyten, K., and Coninx, K., 2011. A unified scalable model of user localisation with uncertainty awareness for large-scale pervasive environments. In *Proceedings of the 5th International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 212–217.
- Aksenov, P., Luyten, K., and Coninx, K., 2011. A unified approach to uncertainty-aware ubiquitous localisation of mobile users. *International Journal of Information Technology and Web Engineering,* 6(4), 20–34.

This chapter presents an approach that models the properties of localisation systems and uses this model to build a unified view on localisation throughout a large-scale pervasive environment. The approach takes into account any localisation system in the environment and requires a semantic description of each system's infrastructure. We use one shared format for exchanging location data, and we explicitly consider uncertain and incomplete information about location in our model and establish its further use in pervasive location-aware applications. We discuss technical considerations, issues and challenges of our approach.

4.1. Diversity in location sensing

Different localisation systems expose different sets of parameters, producing location data in their own format and structure. For example, an object's 2D coordinates can be provided every ten seconds or only be known if the object is within proximity to a certain sensor, making the time of the next update unknown (e.g., a passive RFID tag is detected only if it is close enough to an RFID reader). So that in general, the produced location data would differ in a number of characteristics, such as the level of granularity, frequency of updates, reliability, coverage and availability, means of data delivery, etc. (see Figure 4.1).

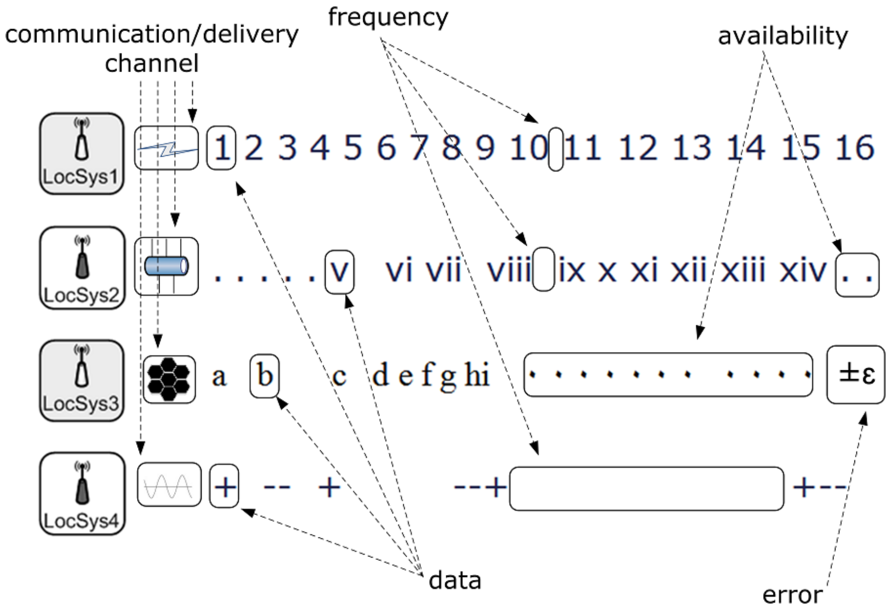


Figure 4.1. Location information provided by a particular localisation system is often organised in a specific format according to this system's needs and restrictions.

The resulting diversity can significantly affect applications that want to make use of different tracking technologies. For example, an existing framework might have little or no support of certain subtle but yet meaningful differences in location data provided by different localisation systems, or it may just not recognise an available system at all (e.g., due to the incompatibility of the communication channel used to receive the data). Besides, the technical limitations and constraints inevitably present in location determination result in uncertainty about location, raising further discrepancies in an application's functionality (Girardin, 2007).

From the user viewpoint, changes, which are in our case initiated by run-time localisation, should be handled in a way that would preserve the continuity of this user's interaction with the application (Massink and Faconti, 2002). And in order to integrate the above limitations and changes of location into the application's behaviour, we need a unified view on the data different localisation techniques provide, as well as on the metadata describing the techniques themselves.

4.2. Location gathering and processing: a unified view

We present an approach that models the sets of properties exposed by various localisation systems. Our goal is to address a comprehensive collection of (types of) included properties in order to cover most common localisation systems, and at the same time to provide a unified view on localisation (orthogonal to variation in localisation systems). This serves two purposes: firstly, the unified model allows application builders to appraise different means of localisation and choose the best fit according to the context of use. Secondly, the model also provides the necessary information to inform the end-user on the behaviour of the localisation system, thereby increasing awareness.

The approach stands upon two building blocks: a unifying component and a location processing component (LPC). The unifying component receives location updates, prepares them for further processing by converting into a common format of location data, and sends them to the location processing component. The location processing component then processes each received update in accordance with the involved reasoning rules. Targeting pervasive environments, we also rely on the fact that internet-capable mobile devices have become widespread and it is now possible to connect to the web almost everywhere in urban areas. Therefore we also unify the means by which information is exchanged between the components and the devices: we use Web to Peer (W2P), a message-oriented peer-to-peer communication system based on HTTP (Vanderhulst, 2010, chapter 4). Thus, this HTTP-based communication framework ensures easy and stable delivery of location updates, and its peer-to-peer topology allows us to decentralise access to location tracking provision, which is consistent with our aim to use both local systems (e.g., Ubisense) and global services (e.g., Skyhook[20]).

The structure of our system is outlined in Figure 4.2, and the components' concepts are discussed in detail in sections 4.3 and 4.4, respectively.

4.3. The unifying component

As noted, the unifying component's main goal is to pre-process diverse location sensor readings with respect to the data they deliver, so that they are treated equally within the environment.

4.3.1. Global Sensor Networks (GSN)

To address the diversity, we introduce a software layer that converts different

[20] http://www.skyhookwireless.com

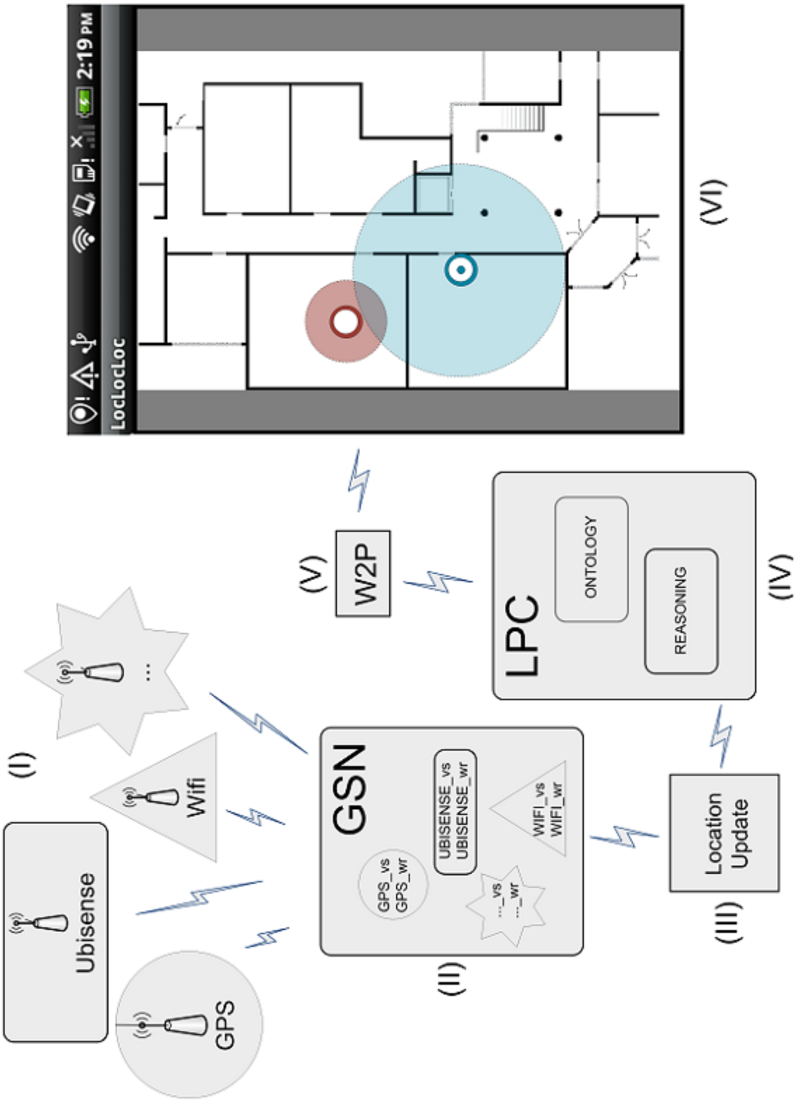


Figure 4.2. System overview: Global Sensor Networks (GSN) (Aberer et al., 2007) (II) handles all incoming location data (I) and produces a location update in the unified format (III).The location processing component LPC (IV) processes the update and sends it to the interested applications (VI). All communication in the environment happens via Vanderhulst's (2010, chapter 4) W2P protocol (V).

formats of location tracking data into a common, unified format. The layer employs the Global Sensor Networks (GSN) platform, a middleware for diverse sensor data processing, proposed by Aberer, Hauswirth, and Salehi (2007).

Why GSN?

The idea of collecting and managing diverse sensor readings within a dedicated system is not new, and attempts in this field have yielded a number of solutions. Aberer et al. (2007) themselves discuss (section VI) the main differences GSN has in this regard to other systems for dealing with sensor networks, such as Shneidman et al.'s (2004) Hourglass (with a focus on "maintaining quality of service in the presence of disconnections"), Franklin et al.'s (2005) HiFi (targeting static environments), or Gibbons et al.'s (2003) IrisNet (with a two-tier architecture for sensing and for storing data, respectively). In our case, we opted for GSN due to a number of different reasons, which all together made it the fittest choice at the time:

- GSN provides a general purpose infrastructure, i.e. it is not bound by application specifics, coordinate systems, spatial models, etc.;
- GSN supports introducing new (location) sensing techniques at run time, i.e. is applicable to dynamic environments, and supports plug-and-play detection and deployment;
- GSN takes a (symmetric) peer-to-peer perspective, i.e. it is applicable to large-scale environments;

Besides, the important practical aspects were its implementation in Java and readily available source-code supported by detailed documentation, which contributed to the time-factor. And last but not least, GSN had proved to work successfully in multiple occasions of real-life actual sensor network set-ups as was demonstrated in its statistics and reports[21], which contributed to its trustworthiness.

It is important to note that we do not consider the actual location processing and reasoning activities at this stage. The unification of location data should not be confused with some existing solutions for location sensing data management, such as Ranganathan et al.'s (2004) MiddleWhere or Hightower et al.'s (2002) Location Stack projects, which might look similar at the first occasion. The unifying process should rather be seen as a complementary intermediate step.

[21] http://sourceforge.net/apps/trac/gsn/ [as by 13 Mar 2012]

GSN organisation

Figure 4.3 shows a schematic representation of the GSN physical organisation. In general, sensor readings from a sensor network arrive to a dedicated sink node, which is connected to a computer where an instance of GSN is running. Base computers may in turn be connected into a network, so that GSN instances responsible for sensors at different sensor networks (and at different locations, in particular) are able to communicate and share sensor data between each other via a GSN-to-GSN communication protocol in a peer-to-peer style using standard Internet and Web Services protocols. There are no assumptions on the structure of sensor networks; the only requirement is that the sink node communicates with the base computer via a software component (wrapper) conforming to the GSN API.

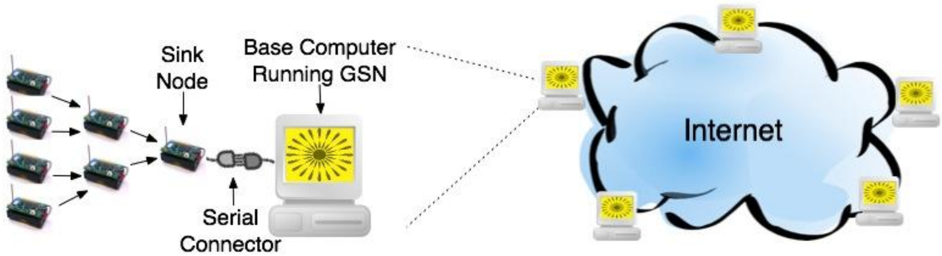


Figure 4.3. GSN model, reproduced from (http://sourceforge.net/apps/trac/gsn/wiki/Introduction) [accessed 20 Feb 2012].

Receiving data in GSN. Java-based wrappers

GSN is able to receive sensor data in two ways: when the data are sent by the source (event-based) and when the data are asked for by GSN itself (polling-based). The received data are processed by a wrapper, which is a Java class extending GSN's `AbstractWrapper` parent class. The wrapper converts the received data into a specific data model, forming a so called `StreamElement` object. In general, data from each source may arrive via a different channel and protocol. In order to standardise the data delivery channel for an arbitrary data source, we additionally introduced a pre-processing step within the wrapper, which converts the data received by the wrapper into a W2P `Message` object. It allows the `StreamElement` object to be addressed the same way, using always the same W2P message format for specifying which data to pass to the `StreamElement` object. Listing 4.1 shows an example of this procedure. In this notation, if a localisation system does not have a certain component (e.g., there is no update rate for passive RFID tags), the corresponding field should be left empty when assembling the `StreamElement` object, and it will be treated appropriately during the location processing stage.

```
postStreamElement(
        w2p.getLocationMessage().getTimestamp(),
        w2p.getLocationMessage().getTagId(),
        w2p.getLocationMessage().getLocSysId(),
        w2p.getLocationMessage().getUpdateRate(),
        w2p.getLocationMessage().getX(),
        w2p.getLocationMessage().getY(),
        w2p.getLocationMessage().getError(),
        ...
        ...
);
```

Listing 4.1. An extract from a sample Java-based wrapper required by GSN.

Also, each wrapper can be assigned a simple name in the GSN's "`conf/wrappers.properties`" configuration file, e.g., as Listing 4.2 shows.

```
...
#Location sensing wrappers
ubisense=gsn.wrappers.locsys.UbisenseWrapper
wifi=gsn.wrappers.locsys.WifiWrapper
gps=gsn.wrappers.locsys.GpsWrapper
...
```

Listing 4.2. A wrapper may be assigned a short name, for the simplicity of referencing.

This simplification purposes to ease future referencing to the localisation system's input data stream. One does not need to be concerned with where the up and running localisation system belongs to or what it uses for processing incoming data; one should only specify the wrapper's custom short name when defining the second key component in GSN, the virtual sensor, which processes the data acquired by the wrapper.

Processing data in GSN. Virtual sensors

A virtual sensor (VS) is an abstraction that corresponds to a data stream received from a data producer of any type (e.g., a sensor itself, a video camera, etc.) or from other VSs. One VS may contain several input streams but always one output stream, generated in accordance with a processing logic defined in this VS. A VS is declared in XML, which contains a description of the required details and configuration for its use, including, among others, metadata for identification and discovery, the structure of the output data stream, and the name of the processing class to process the received data. For example, Listing 4.3 shows a VS that processes location updates received from an instance of Ubisense.

```
1  <virtual-sensor name="ubisense-edm" priority="10">
2      <processing-class>
3              <class-name>gsn.vsensor.W2PVirtualSensor</class-name>
4              <init-params />
5              <output-structure>
6                      <field name="timestamp" type="varChar(13)"/>
7                      <field name="tag_id" type="varChar(15)"/>
8                      <field name="locsys_id" type="varChar(8)"/>
9                      <field name="update_rate" type="Integer"/>
10                     <field name="xcoord" type="double"/>
11                     <field name="ycoord" type="double"/>
12                     <field name="error" type="double"/>
13             </output-structure>
14     </processing-class>

15     <description> Ubisense EDM </description>
16     <life-cycle pool-size="10" />
17     <addressing />
18     <storage history-size="1000000" />

19     <streams>
20             <stream name="input1">
21                     <source alias="source1" storage-size="1" sampling-rate="1">
22                             <address wrapper="ubisense" >
23                                     <predicate key="port">8888</predicate>
24                             </address>
25                             <query> select * from wrapper </query>
26                     </source>
27                     <query> select * from source1 </query>
28             </stream>
29     </streams>
30 </virtual-sensor>
```

Listing 4.3. A virtual sensor for processing location data received from Ubisense.

Here, the output structure (lines 5-13) describes a localisation system's metadata and the location data produced by this system. The value "ubisense" assigned to attribute "wrapper" of the address of the input stream's source (line 22) indicates that the data are to be accessed from a local wrapper with name "ubisense" (see also Listing 4.2). If the data are to be obtained from another virtual sensor of another GSN server, the "wrapper" attribute is assigned the reserved keyword "remote". As stated, the architecture of the GSN framework supports peer-to-peer communication between multiple GSN nodes in the environment; therefore location data provided by a localisation system attached to a remote GSN node are accessible through the network, thereby ensuring the required scalability with respect to location processing. The received and conformed data are finally taken over by a processing class for this VS. For our unified format, we developed a dedicated processing class `W2PVirtualSensor` (line 3). It receives the processed output stream location data, arranged in accordance with the specified output structure, and sends them as a location update for further processing and reasoning by the Location Processing Component (see section 4.4).

4.3.2. Discussion

Both the virtual sensor and the Java-based wrapper have to be created at the design phase of each localisation system and then made available when the system is activated in the environment. Their creation is a one-time activity and does not influence the system's performance at run-time. Nevertheless, the creation requires an acquaintance with the Java programming language and the XML notation. This may become a barrier to the introduction of a custom localisation system to the environment. To aid in this inclusion, our template wrapper already includes the W2P communication set up for the case of W2P-based delivery channel. Using this wrapper, the creation of a particular Java-based wrapper would only require that the data the localisation system in question delivers are described in accordance with the VS' output stream structure format. If the initial delivery channel is different (e.g., UDP), GSN provides an implementation of wrappers for a number of protocols (e.g., HTTP, UDP). For example, in order to receive UDP messages from Ubisense, the provided UDP wrapper was extended with a W2P initialisation block and was modified to conform to the format of Ubisense UDP messages, in accordance with the output structure defined in the VS for Ubisense. Steps of this kind are inevitable, for it is the creator of the localisation system who knows what location data this system provides, and in which order.

This section explained the unifying component of our unified approach to coping with location sensing data. The aim of the component is three-fold: 1) to acquire location data from location providers, 2) to prepare the data for further processing by converting into a unified format that also conforms to the involved HTTP-based communication channel W2P (see Figure 4.2(III)), and, finally, 3) to send the unified data to the location processing component, whose functionality is explained in the next section.

4.4. The Location Processing Component

The location processing component (LPC) (see Figure 4.2(IV)) is a custom application that processes incoming location updates and decides about the locations and statuses of the tracked objects.

4.4.1. Ontology of localisation systems and their data

At the core of LPC is an ontology that describes the semantics of localisation systems and of the data they produce. As with the ontology of spatial relationships introduced earlier in section 3.4.4, our localisation ontology adheres to the Web Ontology Language (OWL) that has been discussed in section 3.4.2.

The ontology has two main concepts: the semantics of a localisation system itself and a description of the location data this system produces. Figure 4.4 shows an example of the localisation ontology visualised as a graph.

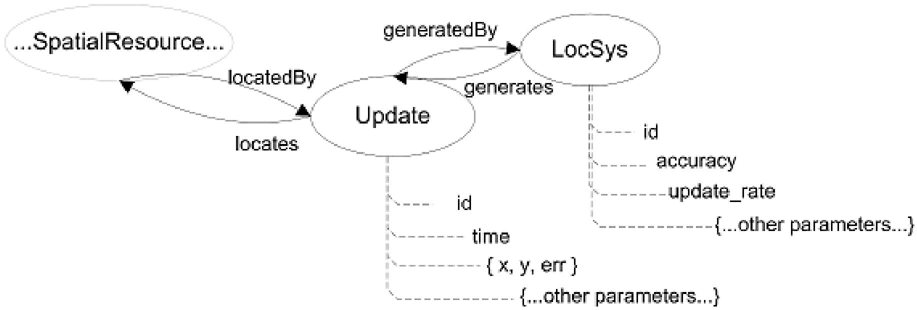


Figure 4.4. An example of an ontology for modelling localisation, with separate concepts for describing the properties of a localisation system and the details of a single location update, i.e. the location data produced by the system.

Location updates of the localisation ontology connect to spatial resources in the spatial ontology by establishing a "locates/locatedBy" relationship pair between an instance of `Update` and an instance of `SpatialResource`. In general, a single object may be located by several localisation systems, so that an instance of `SpatialResource` may have several `Update` instances locating it. There are two cases when it is possible: a) a spatial resource (a user, a device) is located by several localisation systems, e.g., a Wifi access point-based proximity localisation system and a GPS fix; b) a user owns several devices (e.g., a laptop and a smartphone), each of which is located separately. Using these location data all together, the location processing component applies a location solution algorithm to update the `Location` of the corresponding `SpatialResource` (see Figure 4.5).

A localisation system's "virtual sensor" (see section 4.3.1) describes, using XML, two categories of information. The first one contains description information such as this system's id, frequency of updates (update rate), accuracy if announced, and where in the environment it belongs; as well as a location measurement's descriptions, such as its tag's id and a timestamp. The second category is the structure of the actual location data, according to the specification provided in the virtual sensor at this system's design phase. When a processed location measurement arrives to LPC as a W2P message of the unified format, the corresponding `LocSys` instance is looked up in the localisation ontology (or a new instance is created for a new localisation system with the received metadata).

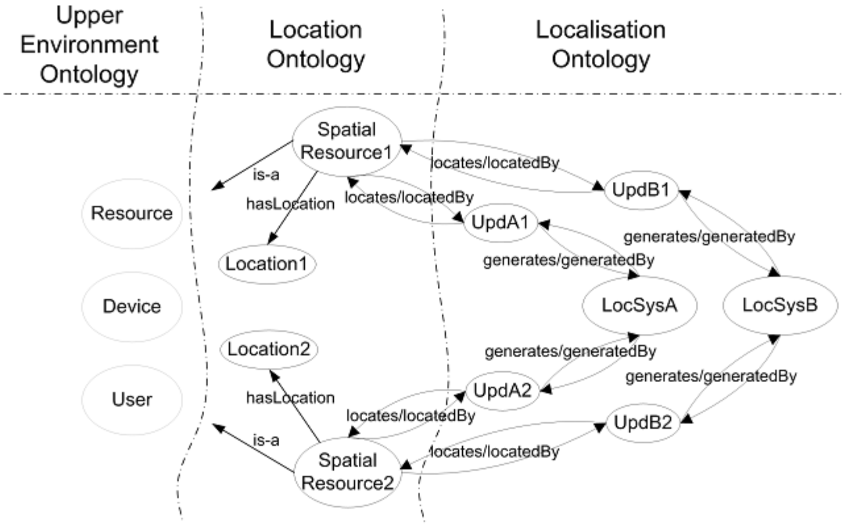


Figure 4.5. A spatial resource may be located by one or more localisation systems. Using the location data received from all localisation systems, the location processing component updates the location of the corresponding spatial resource.

The fields that belong to the measurement and contain an update on a spatial resource's location are stored in the corresponding `Update` instance, as appropriate (an existing one is updated with new location data or a new instance is created). As mentioned earlier in section 4.3.1, if a localisation system provides no information about a field, then this field is left empty in the received location update and is therefore assigned no value in the result. This way, the XML-based virtual sensor defined in GSN acts as a protocol for registering a new localisation system and notifying about its updates, and the ontology described using OWL handles the details of various localisation systems and the location updates they produce.

4.4.2. Current processing scheme

We noted in section 1.3, p.8, that the current work did not intend to elaborate on the algorithmic aspects of experiences of working with multiple localisation technologies. Such concepts as fusing location data provided by several localisation systems, improving the accuracy of localisation, estimating an error of the provided measurements, solving location conflicts, as well as various others, are the topics of separate investigations with many examples of dedicated work (e.g., Hii and Zaslavsky, 2005; Dearman et al., 2007; Aparicio et al., 2008; Niu and Kay, 2008a; Lemelson et al., 2009; Widyawan et al., 2011).

Instead, the location processing component is not bound by any particular localisation technology or approach to location evaluation and therefore should rather be seen as an extensible pool able to incorporate any existing solution for aspects of the above and similar topics. The inclusion of a specific solution would rather depend on its suitability to the needs of the application in question. For example, if we know that one same object is being located by both an acoustic and a Wifi-based localisation system, then Hii and Zaslavsky's (2005) approach may be used for fusing the two systems' location data; and if Bluetooth and Wifi-based localisations are simultaneously available for an object, then Aparicio et al.'s (2008) approach to fusing them should probably be used. The error of a GSM-based positioning can be estimated using Dearman et al.'s (2007) algorithm, whereas for Wifi-based fingerprinting, Lemelson et al.'s (2009) approach suits better. Similarly, if we are able to involve a building ontology (i.e. a familiar environment) into the reasoning scheme, location sensing conflicts, if occur, may be solved by Niu and Kay's (2008a) approach, whereas Widyawan et al.'s (2011) mechanism should better be used for helping one navigate in an unknown environment.

With the above considerations in mind, we nevertheless had to create acceptable conditions for the evaluation of our approach and system in a real-life setting. Therefore, in particular, we still were in need of a reasonably accurate localisation technique in the test environment areas uncovered by the off-the-shelf or set-up localisation technologies also involved in experimenting, such as GPS or Ubisense. A possible workaround could have been to fake location determination over those areas and use a wizard-of-oz approach (see section 5.4.1 for its explanation); but the spatially distributed nature of the planned validation and the run-time reliable simultaneous control of a number of parameters for each tracked object (at least frequency and errors of measurements), together with a lack of human resources to assure for the entire evaluation period, made this approach unfeasible in the long run. Therefore we decided to implement an actually functioning localisation that would prove to suit our planned evaluation scenarios.

Wifi-based localisation

We opted for wifi-based localisation discussed in section 2.3.1, p.21. Our initial preliminary exploration of the Wifi-samples' behaviour revealed that the scans taken at the same location in our test environment varied considerably, being affected by the busy nature of the public space as well as by many wifi-enabled devices used throughout the area on a regular basis. Taking into account these observations, as well as the estimated effort required to record a sufficient fingerprinting map for the area in question, we looked for an alternative that would be more resistant to the above variability, while still having an acceptable accuracy. Here, losing the accuracy was an expected trade-off.

Weighted centroid algorithm The chosen approach to localisation is based on a weighted centroid localisation approach roughly borrowed from (Reichenbach and Timmermann, 2006). This algorithm computes an unknown location of a wifi-enabled device by weighted averaging of the locations of the access points heard by the device at the given location. The locations of the access points are assumed to be known during the configuration phase and are related to an external, either absolute or relative, mapping. Reichenbach and Timmermann (2006) themselves used only four beacons placed in the corners of an area of 300x300cm and experimented with position estimation within the area; Cheng et al. (2005) anchored locations of the access points over a city-scale setup to their GPS coordinates. In our case, we mapped the locations of the access points onto a relative coordinate system that we introduced and associated with the building where the access points were installed. The averaging is expressed using the following formula:

$$L_c = \frac{\sum_{i=1}^{n}\left(w(AP_i)\cdot L(AP_i)\right)}{\sum_{i=1}^{n} w(AP_i)}$$

where L_c is the calculated location of the centroid, $L(AP_i)$ is the location of the *i-th* access point in the scan, and $w(AP_i)$ is the weight of this access point with respect to the device. The weight is based on the heard signal strength and is expressed as

$$w(AP_i) = \frac{1}{(-RSSI_i - 40)^2}$$

where $RSSI_i$ is the signal strength received by the device from the *i-th* access point, and 40 is the empirically determined maximum signal strength in the setup +1. In general, with that manually assigned maximum value, the localisation becomes vulnerable to changes in the Wifi-infrastructure. But setting a stronger, and thus a uniformly "division-by-0"-safe value (e.g., -20dBm can be very rarely seen), would have in this case affected the contribution of the strongest access points as well, making the centroid location closer to the arithmetic average, hence less precise and accurate. In free space, the strength of a radio wave follows the inverse square law with distance, i.e. it is inversely proportional to the squared distance from the wave source. In environments with obstructions, such as buildings and urban areas, the optimal value n of this inversion, which thus stands for a measure of the influence of obstructions, is quite sensitive to the environment infrastructure (e.g., Shen et al., 2005; Bose and Foh, 2007), but it still remains an indicator of distance, though sometimes weak. Since we did not intend to determine the optimal value for our set up, we let n=2 and so made each $w(AP_i)$ be a distance indicator from the corresponding AP_i, which is applicable at longer distances, too (Cheng et al., 2005). By accumulating all heard signals in this manner in the above formula for L_c, we were able to get acceptable location estimates for the area in question.

```
Location calculateCentroid(ArrayList<AccessPoint> wifiScan) {
        Location l = new Location ();

        double x=0.0,y=0.0;
        double totalWeight = 0.0;
        double d = 0.0;

        //calculate centroid location
        for (AccessPoint ap : wifiScan) {
                x = x + weight(ap) * getLocation(ap).getX();
                y = y + weight(ap) * getLocation(ap).getY();
                totalWeight += weight(ap);
        }

        l.setLocation(x/totalWeight, y/totalWeight);

        //estimate error
        for (AccessPoint ap : wifiScan) {
                d = d + findDistance(l,ap);
        }

        l.serError(d/(double)wifiScan.size());

        return l;
}
```

Listing 4.4. Calculation of the weighted centroid location and its error estimation.

The best candidate set algorithm for estimating the error of a fingerprinting 802.11-based localisation (Lemelson et al., 2009) comprises three steps: 1) select the k best estimates as determined by the positioning algorithm, 2) compute the distances between the best estimate and the other (k–1) ones, and 3) return the averaged distance as the estimated error. Lemelson et al. (2009) reported that in their experimental environment the algorithm achieved the best performance for k=3, and the increase of k resulted in more conservative estimates, affecting the overall performance.

We used the described approach with the difference that we computed the distances from the centroid location to the locations of the access points heard at a given location (there were on average five access points in each sample). An evaluation showed that the accuracy happened to be approximately between 10 and 25 metres in the bigger test environment (see section 5.4) and approximately between 3 and 10 metres in the smaller test environment (see section 5.3).

We note that generalisation of the proposed localisation to other environments may not be straightforward. The calculated weights and the best candidate set algorithm were applied to a moving object's centroid-based, i.e. not a fingerprint, location and without strictly controlling the number of visible access points participating in the error estimation. But as we emphasised earlier, our primary goal is to abstract from any particular localisation technology and consider them as they are, i.e. without introducing any adjustments or improvements. Therefore the developed system's acceptance criterion was its ability to locate the user throughout the areas with a reasonable location error, which it did.

Reasoning

Handover A practical aspect of processing the sensed location data included taking care of the so called handover situations, in which the active tracking changed. For example, when one walks in from the outdoors, their wifi-based localisation would start converging while GPS signals would disappear. Besides, during the test runs of our wifi-based localisation, we discovered that GPS was also sometimes available indoors through the glass roof or near some windows. Since this behaviour was pretty unstable, we had to analyse such situations so as to avoid unnecessary occasional switches.

Hansen et al. (2009) compared several approaches to the handover between GPS and Wifi fingerprinting localisations and concluded that the preference of GPS upon its continuous readings (at least 5 seconds) was the best choice for the environment in question. However, they did not consider positioning errors. Therefore our handover mechanism partially used Hansen et al.'s (2009) results and considered the following aspects: a) a localisation system's reliability measure, i.e. the system had to produce at least two continuous readings in order to become a candidate for the handover; b) system's accuracy, i.e. if both systems were considered reliable in terms of aspect a), then the one with a better accuracy was set as the active. This approach successfully handled both cases: the unexpected case of occasional GPS signals indoors and the expected case of entering and leaving the building intentionally.

Location processing We involved a basic set of parameters, such as the update rate and the positioning error, using which we were able to realise our system's basic functional model taking the uncertainty into account. We then used the reported measurement error and the age of each received location update to decide about this update's validity and appropriateness. The age depended on the update rate, so that a location update was considered obsolete when it became older than either the time of the two next scheduled updates or a context dependent threshold.

In general, reasoning about the validity and appropriateness of a location measurement involves a number of aspects. We have already discussed at the start of this section, p.73, such concepts as preferred mechanisms for location fusion or error estimation, or the availability of a containment ontology (such as a building ontology). Similarly, the heterogeneity of a pervasive environment calls for the inclusion of concepts from other context ontologies, such as a user profile (e.g., the walking speed), a task model (e.g., attending a meeting), or the topology of the environment (e.g., locations on water are invalid). Therefore the LPC's reasoning functionality can be extended with external ontology handling and reasoning solutions. Examples include Ye et al.'s (2007a) unified semantics space model with its containment, adjacency, and connectedness relationships and a corresponding API for reasoning (Stevenson et al., 2010), or Niu and Kay's (2008b) PECO ontology of a building and its dedicated ONCOR reasoner (Kay, Niu, and Carmichael, 2007).

4.5. Summary

This section presented an approach to coping with the diversity of available techniques for location sensing in a pervasive environment. The proposed framework used the provided semantic descriptions of different localisation systems to build an ontology for modelling location sensing process. The key aspect of the approach was the conversion of an arbitrary location update into a unified description format with the help of GSN, a third-party software layer for working with diverse sensor data. The unified update was then processed by a location processing component taking into account the uncertainty the update in question contained. The use of W2P, an HTTP-based message-oriented framework for information exchange, together with GSN's peer-to-peer approach to data collection, ensured the applicability of the approach to large-scale environments.

Part II

Location awareness and relevancy aspects

Chapter 5

Adaptive user awareness

Parts of this chapter have been published as the following journal articles:

- Aksenov, P., Luyten, K., and Coninx, K., 2011. A unified approach to uncertainty-aware ubiquitous localisation of mobile users. *International Journal of Information Technology and Web Engineering,* 6(4), 20–34.
- Aksenov, P., Luyten, K., and Coninx, K., 2012. O brother, where art thou located?: Raising awareness of variability in location tracking for users of location-based pervasive applications. *Journal of Location Based Services*, online version, 23 pages, DOI: 10.1080/17489725.2012.682098.

Additionally to the support of the diversity of localisation technologies, an essential aspect in the creation of location-aware applications for pervasive environments is to make users aware of the variability of their frequently changing location conditions, so that changes are perceived appropriately. This chapter shows how the variability of location context can be visualised intelligently, supported by two user studies that investigate and analyse user needs regarding awareness of the variability of location context.

5.1. Aspects of user awareness and adaptation

5.1.1. Visualising uncertainty

Benford et al. (2006) discuss the importance of addressing uncertainty about location in users' location-based experiences. They propose five strategies of dealing with it – *remove it, hide it, manage it, reveal it,* and *exploit it*, however conclude also that the choice of the appropriate strategy depends on the context of use and application needs.

In some situations, uncertainty can be predicted and hence incorporated into the application at design-time. For example, Benford et al. (2006) used the positions of the GPS satellites at a specific moment for visualising the likely availability of GPS over a certain area at that moment; in particular, shading all buildings permanently black due to unavailability of GPS indoors at all times. But some factors about uncertainty can only become known when they appear, so that they need to be processed and visualised at run-time using the information at hand. Here, Dearman et al. (2007) explored the variability of the location error of a GSM-based localisation in outdoor map-based navigation. They represented the error as a circle projected on the map and centred at the predicted location. The radius varied depending on the level of confidence that the true position was within that circle (see Figure 5.1(i)). The results revealed that showing the error was beneficial in terms of both the perceived difficulty of completing a location finding task and the time required to do it. Lemelson et al. (2008) conducted a paper-based survey that compared several alternatives to visualising the error of users' position indoors. The respondents were asked to choose the most suitable visualisation in each of a number of scenarios, and the results uncovered a clear preference for the suggested in-map visualisations, such as a coloured circle or area on the map (see Figure 5.1(ii)). Burigat and Chittaro (2011) recently investigated user preferences towards visualising the uncertainty caused by a degraded GPS signal. They compared three approaches: a basic visualisation showing the predicted position only, a circle centred in the predicted position and extending proportionally to the speed of walking, and street segments, i.e. the routes along which the participants could actually walk, coloured also according to the walking speed (see Figure 5.1(iii)). The participants were asked to walk along a path, where unavailability of the GPS signal was simulated in a predefined area, and pinpoint their location on the map upon request in the end. The results of the study revealed that the two uncertainty-aware visualisations were perceived to be beneficial, requiring a lower mental demand for the street-colouring and a lower effort for both. Other ways of uncertainty visualisation may also reflect the nature of the localisation system together with the actual infrastructure of the environment the user is in (e.g., see Figure 5.1(iv)).

As we see, the problem of revealing the uncertainty about location sensing to users has attracted researchers' attention. However, representing uncertainty in the most suitable way is still an open question (Opperman, Broll, Capra, and Benford, 2006). Moreover, a representation that would suit general-purpose applications for non-experts may be even more difficult to find (Patel, Kientz, and Gupta 2010).

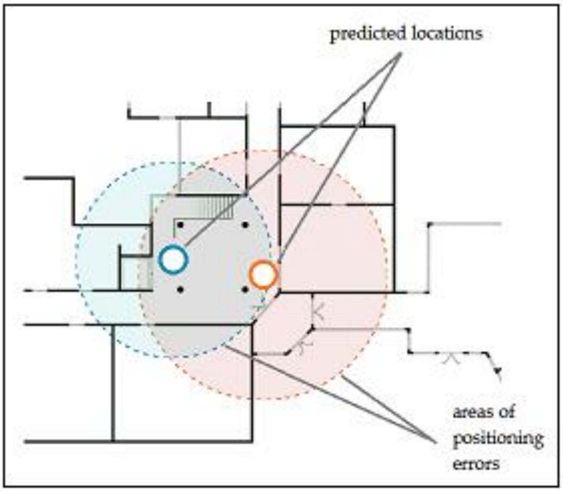
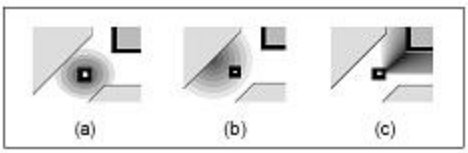


Figure 5.1.

(i) in the circle-based visualisation, a user's predicted location is surrounded by a circular area of the positioning error, corresponding to the accuracy of the location tracking provider;

(ii) in the room-in-map visualisation (figure reproduced from (Lemelson et al., 2008)), the building layout is considered so that the rooms are coloured according to how far they are from the predicted location;

(iii) in the street mode (figure reproduced from (Burigat and Chittaro, 2011)), the streets are coloured according to a user's walking speed, starting from the predicted location;



(iv) error regions may also differ for different sensors: (a) GPS, (b) Infrared-based tracking, (c) an orientation-aware tracking with an accuracy of 90 degrees (figure reproduced from (Baus and Kray, 2002)).

5.1.2. Ontology-powered adaptation of pervasive applications and interfaces

Ontologies have been successfully used to represent the knowledge about domains of interest while allowing developers to reason about this information. For example, Strobbe et al. (2007) presented CASP, a context-aware service platform for combining and organising context information of diverse types with the help of ontologies. They then used their platform to help determine an employee's most likely location by means of reasoning about the statuses and approximate locations of this employee's and their colleagues' personal devices (such as their laptop, PDA, or mobile phone), which were modelled in an ontology. Ongenae et al. (2008) also used CASP to demonstrate improvements in the performance of a hospital's nurse call system after a place-oriented approach had evolved into a person-oriented approach. This approach additionally included reasoning about such information as urgency of a patient's call or a nurse's current status. Cearreta and Garay-Vitoria (2011) addressed adaptation of interfaces of ubiquitous services to users with special needs. They introduced an ontology of users' sensorial and perceptual capabilities and demonstrated how it could be used to determine and generate an appropriate means of interaction with the service automatically. Hervás and Bravo (2011) have shown that the potential of combining knowledge about several domains to provide ontology-powered proactive adaptation of pervasive user interfaces is high. By means of combining domain, device and user ontologies, the authors personalised the information shown to the requestor on a public display at run-time. The evaluation of the proposed run-time personalisation revealed that the required documents were selected in 80.77 percent of the cases.

The applications in the above examples relied on the permanent availability of location tracking. In a pervasive environment, which is usually heterogeneous and dynamic, the behaviour and availability of localisation systems change frequently. Therefore adaptation mechanisms must take this feature of the environment into account as well. In section 5.2, we introduce user interface adaptation that reflects the changes in the localisation conditions using our localisation ontology.

5.1.3. Influence of location-awareness

Location awareness contains and depends on many factors, of which some may matter more than the others, some may be considered either helpful or hurtful, and some will just not make any difference. For example, Dearman, Hawkey, and Inkpen (2005) found out that otherwise helpful awareness about rendezvousing partners' location and movements became detrimental if one of them could not determine what their partner's problem was. Similarly, Lim and

Dey (2011) revealed recently that the same and previously useful awareness became harmful when the certainty of information provided in the same scenario decreased. However, the results were produced by Amazon's Mechanical Turks and thus provided passive judgements only, so these authors still plan to validate their hypotheses in a real-world scenario. Misund et al. (2009) reported that revealing information about locations of other players in a collaborative location-aware chase-and-catch game did not affect the performance in the main task; however, it did make the game more 'fun' to play. Nova et al. (2010) found out that automatic mutual location awareness made the coordination process within the tested group less efficient, as opposed to the groups whose members used self-reporting on their whereabouts.

Thus, understanding the effect and the level of importance of clues on the variability of the location context from the user point of view is useful for developers and designers of pervasive location-aware applications.

5.2. Visualisation component

This section presents the visualisation component of our application. It introduces and explains in detail how such properties as the frequency, the error, and the availability status of a user's location determination can be reflected in the graphical user interface at run-time in order to inform the user about the changes brought in by variability. The component was realised in Java, for smartphones running the Android platform[22].

5.2.1. Visualisation

The examples and the results discussed in section 5.1.1 illustrate that the omnidirectional representation of positional uncertainty as a circular region, which is centred at the predicted position and with the radius corresponding to the error of a position estimate (Figure 5.1(i), p.83), is a popular visualisation that can also be met in many other applications (e.g., Google Maps for mobile[23]). Comparisons of other in-map approaches with the circle-based representation did not reveal preference of any of the experimented (Burigat and Chittaro 2011, Lemelson et al. 2008); whereas it was at the same time preferred to alternative, i.e. not in-map, ways (Lemelson et al. 2008). Therefore we also use a circle-based visualisation. We further extend it to reflect the finer localisation dynamics in situations with several tracking providers in a mixed indoor-outdoor area, and without restrictions on navigating paths. For this purpose, we also consulted

[22] http://www.android.com

[23] http://www.google.com/mobile/maps/

with a professional graphic designer and explained which details and situations we wanted to visualise. The next section describes the resulting extension in detail.

5.2.2. From location tracking to visualisation patterns

Benford et al. (2006, p.122) discern four 'states of being' of a mobile user: *connected and tracked, connected but not tracked, tracked but not connected,* and *neither connected nor tracked*. These states separately consider the tracking of users' location (tracked) and the ability to communicate information, including location, to other mobile users (connected). Based on this classification, we introduce a set of states of awareness about the status of a mobile user's location that reflect the behaviour of an active localisation system and take into account uncertainty. Figure 5.2 shows the four states that read as follows:

(a) a location update is received with an initial positioning error as the circle's radius, and the dot in the centre starts pulsating (see Figure 5.2(a));
(b) the radius extends as a function of the time elapsed since the last update, according to a user's moving speed. If the timeout for receiving the next update has been exceeded, the dot no longer appears (see Figure 5.2(b));
(c) if the second timeout has been exceeded, the position marker turns into a cross (see Figure 5.2(c));
(d) the error area continues to extend until the maximum age for the referred location measurement is reached. After this, the uncertainty area is no longer shown, the cross thickens, and the information about this location is considered outdated and unreliable (the visualisation becomes static) (see Figure 5.2(d)).

Thus, there are several independent elements that contribute to the visualisation in each state – *a dot*, *a circle*, and *a cross*. We will hereafter refer to these elements as *visualisation patterns*. To determine an appropriate visualisation, the status of tracking is mapped onto a subset of these patterns. The decision about the appropriateness of inclusion of each pattern into the visualisation is based on this pattern's validity and importance. The validity is determined by the state of the referred location update, and the importance is either specified by the designer or determined by considering this element in connection with other context models (e.g., privacy settings).

The concepts defined in the ontology of localisation systems that we described in chapter 4 carry a set of common attributes contributing to the unified model of localisation. Some attributes, such as the update rate, characterise a localisation system rather than an individual location measurement. This is done for the sake of correctness of addressing the details of the localisation process and for preserving the actual state of affairs. Such situation, for example, happens when the same system locates several entities and sends independent updates with partially identical information, or when the same entity is located by several systems, so that several independent updates are received as well. During the location processing and reasoning, each parameter becomes part of the unified location update component used for the identification of the appropriate visualisation patterns. In the scope of this work, the following attributes are considered: *the 2D location data, the location error estimate,* and *the update rate,* i.e. how often the location update is produced (by the localisation system).

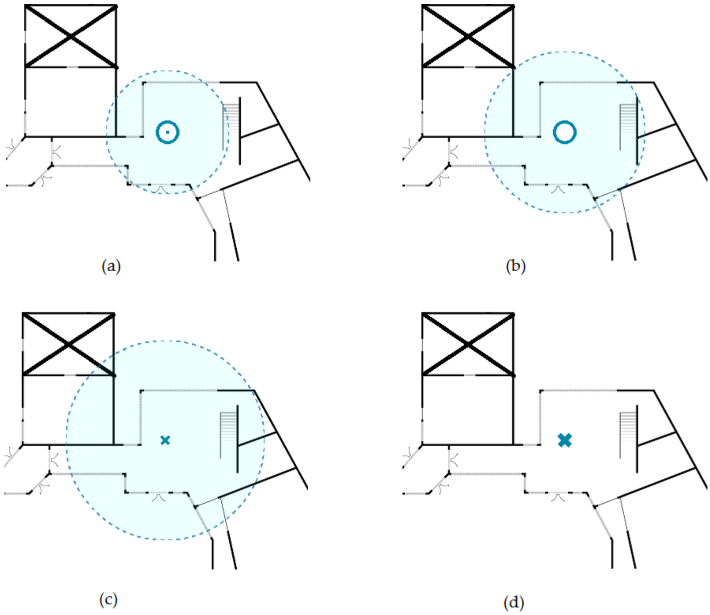


Figure 5.2. Visualisation of the status of a user's location tracking distinguishes four states of awareness: (a) a regular update is received; (b) an update is missing; (c) location tracking is possibly unavailable; (d) location information is outdated (visualisation is static).

The proposed location-driven transformation of the visualisation of a user's position reflects the variability of run-time location sensing in a way that will prepare the users for a possible change in their tracking and the application's behaviour. The visual dynamics of the transition between the states informs the user about the upcoming change and, by making the transition smooth, helps to preserve the continuity of their interaction with the application. We describe in the next section how appropriate visualisations of the users' location states from Figure 5.2 are determined by applying a corresponding rule to the given location tracking conditions.

5.2.3. Ontology-based patterns selection

Figure 5.3 depicts the flowchart of visualisation generation. Three kinds of requests to generate the visualisation are possible: two of them would query the ontology and one would be handled locally within the application.

A request of the first type is triggered when an update (L' in Figure 5.3) on a user's location has been received and needs to be communicated. This update is first processed accordingly by the Location Processing Component (see section 4.4, p.71), which updates the ontology and sends the updated instance to the mapping table (L to UI Mapping Table in Figure 5.3). The mapping table is a collection of rules that determine which subset of visualisation patterns corresponds to the attributes of the received location update. Listing 5.1 shows the set of rules, formed in the Semantic Web Rule Language[24] (SWRL) in accordance with the concepts of the localisation ontology and the assumptions we made that show how a user's location tracking details are used to infer the appropriate visualisation condition from the UI Mapping Table in Figure 5.3. The mapping

```
00 //common part
01 Resource(?r) ^
02 hasId(?r,?id) ^
03 locatedBy(?r,?upd) ^
04 error(?upd,?e) ^
05 generatedBy(?upd,?ls) ^
06 hasRate(?ls,?rate) ^
07 hasTime(?upd,?time) ^
08 locsys:age(?age,?time) ^
09 locsys:timeout(?to,?rate) ^

1.10 //condition 1
1.11 swrlb:lessThan(?age,?to))
1.12 => locsys:disseminate(?id,xsd:string("condition1"))

2.10 //condition 2
2.11 swrlb:greaterThanOrEqual(?age,?to)) ^
2.12 swrlb:lessThan(?age,op:numeric-multiply(?to,xsd:int(2)))
2.13 => locsys:disseminate(?id,xsd:string("condition2"))

3.10 //condition 3
3.11 swrlb:greaterThanOrEqual(?age,op:numeric-multiply(?to,xsd:int(2))) ^
3.12 swrlb:lessThan(?age,op:numeric-multiply(?to,xsd:int(4)))
3.13 => locsys:disseminate(?id,xsd:string("condition3"))

4.10 //condition 4
4.11 swrlb:greaterThanOrEqual(?age,op:numeric-multiply(?to,xsd:int(4)))
4.12 => locsys:disseminate(?id,xsd:string("condition4"))
```

Listing 5.1. A combined representation of the SWRL rules that use the localisation ontology to infer about a corresponding resource's appropriate visualisation condition based on this resource's location tracking status.

[24] http://www.w3.org/Submission/SWRL/

table then assembles the appropriate subset (UI_SET in Figure 5.3) of visualisation patterns corresponding to the inferred condition and sends it to the application that eventually adds the corresponding visualisation to the user interface. For example, in Figure 5.3, the generation of the visualisation for a newly received update (case L', t=0) is illustrated, so that the UI_SET receives the {P+,C(R+tV)} combination of the visualisation patterns to be visualised.

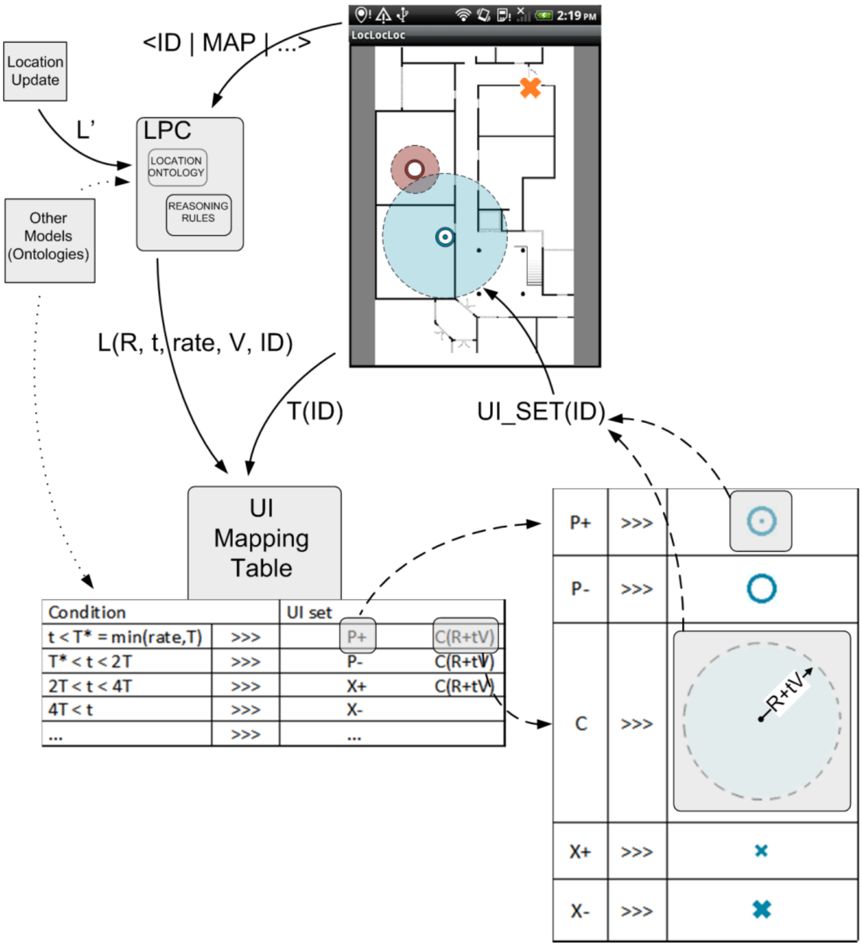


Figure 5.3. The flowchart of visualisation generation. Depending on the type of the request – a new location update (case L'), a previously inactive/not-shown user (case <ID | MAP |. . .>), or a timeout (case T(ID)) – a set of the appropriate visualisation patterns, UI_SET(ID), is formed according to the details of the request.

A mapping request of the second type queries the ontology for the available details on location and state of some particular user or a group of users. In this case, LPC processes the corresponding query and sends the results to the mapping table that does the selection. For example, the query in Listing 5.2 is generated if the user of a client application chooses to look at the map of an area with id ‘main_building’. Furthermore, for each resource thus selected, its visualization is determined by one of the above SWRL rules.

```
PREFIX lo: <http://site.org/locsys#>
SELECT ?id ?x ?y ?err ?time ?rate
WHERE
{
?lu a lo:LocationUpdate ;
    lo:activeMap 'main_building' ;
    lo:locates ?id ;
    lo:x ?x ; lo:y ?y ; lo:error ?err ;
    lo:time ?time ;
    lo:generatedBy ?ls .
?ls lo:rate ?rate .
}
```

Listing 5.2. A sample SPARQL query that asks the localisation ontology for location tracking details of all users who belong to the specified map (‘main_building’).

Finally, a mapping request of the third type is activated in the case when there are no updates on the location tracking of a currently visualised user. The application sends this user’s attributes (T in Figure 5.3) to the mapping table that determines a degraded version of the visualisation. Here, the degradation simply shifts the active condition from the currently applied to the lower one according to the mapping table hierarchy (i.e. no need to involve the ontology, for there have been no updates to the location of the user in question).

We would like to note here that all improvements and decisions on the quality and validity of the provided update are entirely in the hands of LPC. The extensions on the visual awareness introduced here do not influence the availability or the quality of localisation itself – they only reveal its details, existing problems and limitations to the user and in accordance with this user’s needs.

5.2.4. Map support and location overlays

We used Google Maps to show both outdoor and indoor locations (Figure 5.4(left)). The options menu allowed the smartphone user to see the locations of the tracked users on the smartphone's screen or to zoom in to the user's own location.

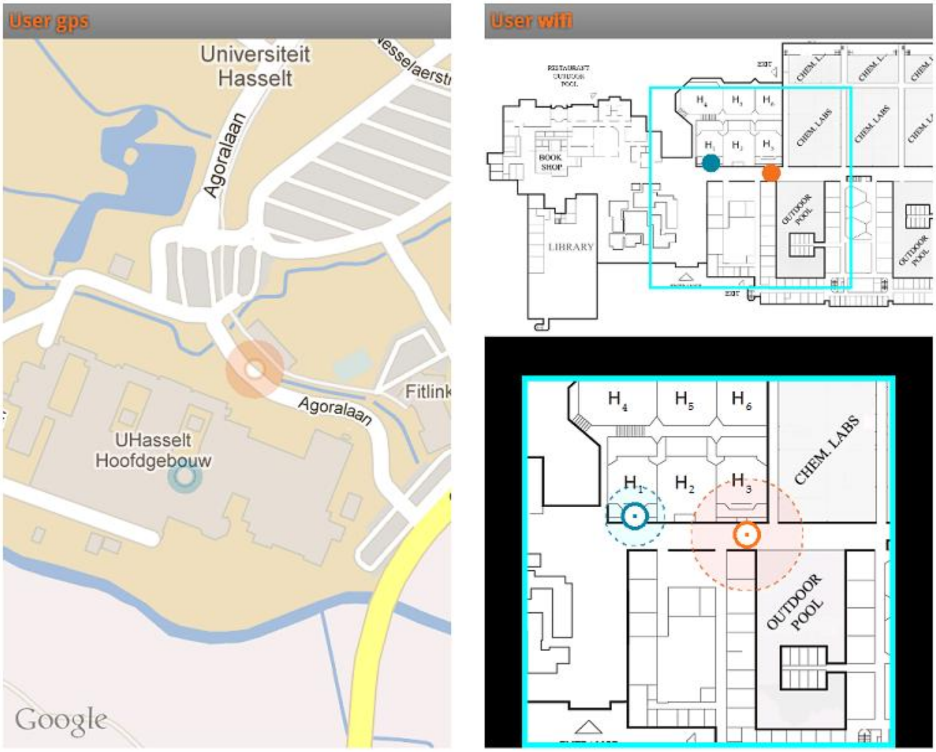


Figure 5.4. The application shows the locations of both the participant and the target in the corresponding colours on a map. When zoomed in further in the proximity of the indoor area, a map of the building layout in finer details is displayed.

When zoomed in further beyond the Google Maps maximum zoom-level, an option some users may need in the proximities of indoor environments in order to explore further details of a particular building, the building layout in finer details was displayed: it showed a map of the entire building and of its zoomed-in segment, where the details of the users' locations were visualised in the appropriate colours (see Figure 5.4(right)). While within the provided detailed building map, users could pan both maps at their convenience and zoom out if they wanted to return to Google Maps (Figure 5.4(left)). This simplification of panning to within the building map was justified by the conditions of the experiment that we made for validating the proposed adaptation concept and that is described in section 5.4. The experiment was designed for two users and the building map also covered the proximity of the indoor environment, so that no context with respect to other users was lost. In multi-user cases, where spatially distributed users must be visualised simultaneously within the context of the same screen, the extension of the panning to outside of the detailed building map should in this case normally display the appropriate area of Google Maps at the corresponding zoom-level. We also took into account Seager and Stanton Fraser's (2007) earlier finding, which revealed that the users of map-based navigation applications preferred physical rotation of their mobile device to align the displayed map with their walking direction; so we did not realise automatic rotation.

5.2.5. Notifications

We also introduced two types of notifications. Notifications of the first type inform the user about a change in the active tracking system (see Figure 5.5). The influence of this method of notifying users and its implications are studied in section 5.3 reporting about a dedicated user study.

Figure 5.6 shows the second type of notifications: when a tracked user joined or left the displayed indoor map, a corresponding notification was shown on the screen and this user's visualisation was, respectively, generated or removed.

The next two sections describe and present the results of two real-life field studies, in which we evaluated the effect of the described approach. In use case presented in section 5.3, we investigated users' preferences regarding combining two types of awareness, and the use case described in section 5.4 evaluated and analysed how the proposed dynamic visualisation using visualisation patterns affects navigation strategies of the users of the application while they are trying to chase a moving object.

5.3. Use case 1: Informed changes

This user study aimed at getting users' opinion on whether the awareness of changes in the active location tracking influences the understanding and satisfaction from using the application.

5.3.1. Preparation and setup

In this user study, we switched off the Google Maps support: the experimental area was mainly indoors and the involved outdoor area was fully covered by the custom building map.

Figure 5.5. When the active localisation changed, the application showed an appropriate notification message about this event.

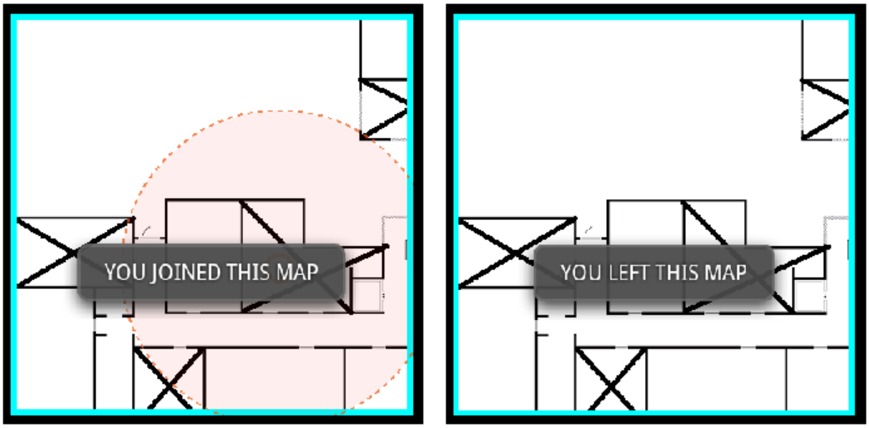


Figure 5.6. An appropriate notification is shown (and the visualisation is updated) if a user joins or leaves the indoor environment.

Localisation and experimental area

We involved three localisation systems: Ubisense, GPS, and a WiFi-based localisation explained in section 4.4.2, p.73. Each system covered a different area (see Figure 5.7) and had a different level of accuracy.

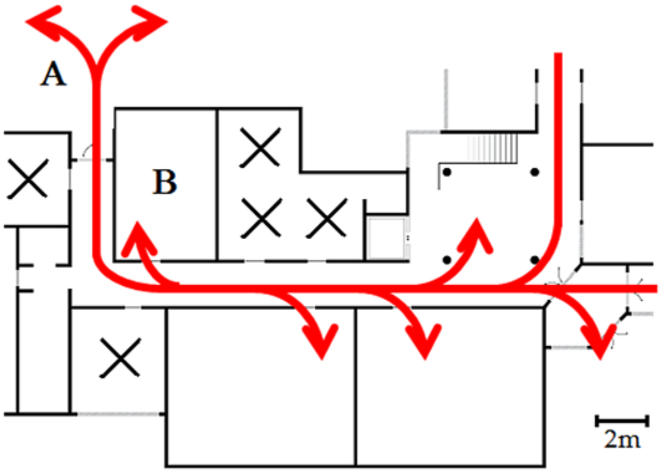


Figure 5.7. A part of the map of the area involved in the user study. Positions around zone A are tracked by GPS, around zone B by Ubisense, and the rest of the area is controlled by a WiFi-based localisation. The areas marked with a cross cannot be walked into. The arrows indicate the available walking paths.

Overlaps existed only in the Ubisense coverage area (room B in Figure 5.7) where the WiFi-based localisation produced a location as well but the preference of the considerably more accurate Ubisense was always straightforward.

Experimental conditions, visualisations and hypotheses

We confined the conditions of the study to a maximum of three moving objects on the map at the same time, and we used a simple colour-based visualisation to distinguish between them. A graphic designer advised us to employ the following three colours: a tint of orange ("F37021" in HEX), a tint of blue ("0084A2" in HEX), and a tint of green ("78BF1C" in HEX). Any of these colours was assigned to each participant for the duration of the experiment, and there were no remarks about any disadvantages or inconveniences caused by the assigned colour. The application was deployed on an HTC Desire Android smartphone.

We varied the awareness mode regarding the detected location: one mode showed the position alone without any additional information whereas the other mode also displayed the positioning error. We also varied how the change of the active tracking system was reflected: in one mode, the user was notified about a change by a notification message from Figure 5.5, p.93, whereas no feedback was provided in the other mode. Each mode of awareness of uncertainty about position was then coupled with each mode of notifications about changes in the active tracking. This resulted in four different modes of presenting the information: (A) both uncertainty and notification; (B) only uncertainty; (C) only notification; (D) none of them. Figure 5.8 shows the application screenshots in each of the four modes, labelled as above.

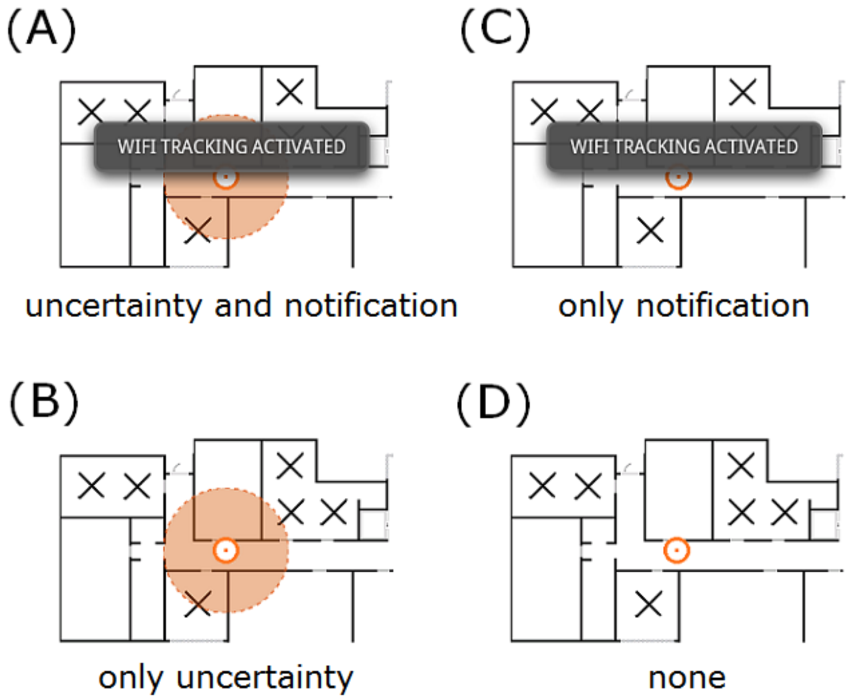


Figure 5.8. Four different modes of awareness of the statuses and changes in a user's localisation were offered to the participants in a user study.

Using these four modes, we wanted to verify the following two hypotheses that we made about awareness:

- H1. Under the same visualisation conditions, users' impressions about being aware of the changes in the active tracking are positive. That is, mode A would be preferred to mode B, and mode C – to mode D.
- H2. Mode A would be ranked as the preferred mode out of the four.

Participants

Eight people (5m, 3f) participated in our study. Seven of them had an IT background and one was a graphics designer. They were all employees, visitors or computer science students of the university, but were not involved in our research. Two participants did not wish to disclose their age, and the other six were between 20 and 34 years old (*M*=26.7, median=27). We asked the participants to rank their expertise in 1) using map-based applications for navigation, and 2) working with smartphones, on a five-point Likert-type scale: 0-none, 1-a bit, 2-some, 3-quite a bit, 4-a lot of. The average values were, respectively, 2.1 (median=2) for navigation experience and 2.6 (median=2.5) for smartphone experience.

Experimental procedure and data collection

We used a within-group design, so each participant evaluated all four application modes. The assignment of the modes per participant was done according to a Latin Square design. Table 5.1 shows the final assignment.

Table 5.1. The order of application modes in which each participant received them.

Participant	Mode (A)	Mode (B)	Mode (C)	Mode (D)
P1	3	4	1	2
P2	1	2	3	4
P3	2	1	4	3
P4	3	4	1	2
P5	4	3	2	1
P6	4	3	2	1
P7	2	1	4	3
P8	1	2	3	4

The task we asked our participants to do in each mode was the same: carrying a smartphone with our application running in one of the four modes, the participants were instructed to walk throughout the area shown in Figure 5.7 (the arrows in the figure indicate the available walking paths). As they walked, the participants had to observe the presented information available in the currently running mode. There were between 1 and 3 objects tracked at every certain moment. The participant was always one of the objects, and the other two objects were played by experimenters. Upon completing the walking part in each mode, the participants were asked to evaluate their experience of using the application in that mode using the following relevant criteria of the NASA-TLX questionnaire[25]: *mental demand, performance, effort*, and *frustration*. We would like to note here that the suggested task did not have any constraints with respect to time or labour; therefore we did not include the other two criteria, *physical demand* and *temporal demand*, present in the original questionnaire, for evaluating them would not have been representative.

[25] http://humansystems.arc.nasa.gov/groups/TLX/

The participants were also invited to provide additional comments to support the given rankings. In the end of the experiment, we also asked the participants to rank all four modes together. Each participant needed on average 40 minutes to complete the entire experiment.

5.3.2. Results

Verification of H1

To verify H1, we compared the TLX scores each participant gave to mode A (uncertainty and notification) with those the same participant gave to mode B (only uncertainty); and the scores given to mode C (only notification) with those given to mode D (neither uncertainty nor notification).

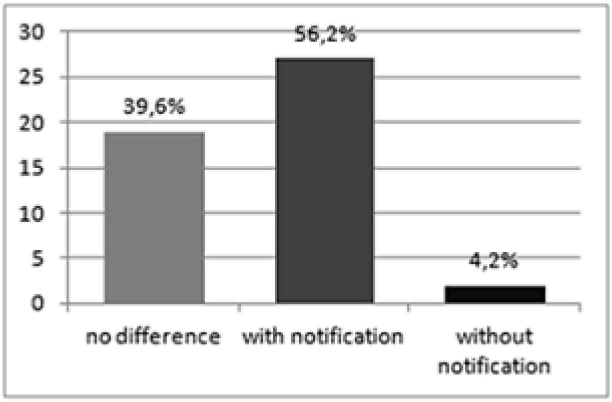


Figure 5.9. An informed switch between location tracking systems (i.e. with an explicit notification) was the preferred option within the same visualisation.

Overall, the participants provided 128 scores (8-participants x 4-TLX-criteria x 4-modes), which formed 64 pairs of values grouped by mode for comparison as appropriate (i.e. {mode A vs. mode B} and {mode C vs. mode D} for each of the four criteria by each participant). P6 and P8 who evaluated altogether 16 pairs experienced no difference between the two modes within either pair. The other participants evaluated altogether 48 pairs, 19 of which were ranked as giving no change, 27 times the informed case was preferred, and 2 times the mode without notifications was ranked higher, all the facts together confirming H1. Figure 5.9 compares these numbers in a chart.

It is worth noting that the average difference of the preference of the informed mode to its non-informed compartment in terms of the 21 gradations of the TLX-scale was much greater (*M*=5.12, *SD*=2.9) compared to those when the non-informed mode was preferred (the two differences were 1 and 2 gradations, respectively). For example, P7 supported the difference in the scores she gave to the required mental demand in modes C (7) and D (18) with a conclusion that *"since no precision information was shown, (I) was thinking which tracking was actually currently active. (I) Should be outside of the Ubisense area but still unknown if (I am) tracked by it or not."*

Verification of H2

Table 5.2 shows the scores the participants gave to each mode in the end of the experiment, from 4 (best) to 1 (worst).

Table 5.2. Scores, from 4 (best) to 1 (worst), that each participant gave to application modes (A), (B), (C), and (D)

Participant	Mode (A)	Mode (B)	Mode (C)	Mode (D)
P1	4	2	3	1
P2	4	3	2	1
P3	4	2	3	1
P4	2	1	4	3
P5	4	3	1	2
P6	4	3	2	1
P7	4	3	2	1
P8	3	4	1	2
Average score	3.63	2.63	2.25	1.5

Six of the eight participants preferred the mode with both uncertainty and notification active during navigation (mode (A)), confirming H2. Participant P4, who preferred mode C (only notification) to the three others, explained his preference by saying, *"The uncertainty circles brought too much information onto the screen, but notification about a change was helpful because it allowed me to know the situation"*, thus voting for the informed visualisation also. Participant P8, on the contrary, found notification messages unnecessary because, as he stated, *"when (I) got notified, (I) started to wonder what the new tracking was about. (I) lost focus and thought of the information that appeared."* However, he found the information on the uncertainty quite helpful. It is interesting to note that P8 was the only one who gave the maximal score (4) to his experience with both map-based navigation and smartphones.

5.3.3. Conclusions and discussion

The main purpose of the study was to collect and evaluate users' impressions about the additional awareness of a diversity of localisation systems. Prior research results confirmed people's preference for receiving information about their positional uncertainty, and we additionally allowed users to know the reasons for this uncertainty. We did not intend to differentiate the degrees of the effect the awareness would have but rather wanted to learn whether there would be a benefit from knowing about it. The results of the evaluation showed users' acceptance and preference for automatic and informed changes in their location tracking conditions, and that they chose to remain aware of the cause of the changes they experienced.

Also, among the comments the participants provided on the use of the visualisation, we identified a common remark coming from several participants independently. They expressed that while the visual notifications turned out to be useful, a less straightforward way of delivering this type of information would be beneficial. We took this observation into account in the second use case, where the id of the tracking system that produced the last known (i.e. the most recently received) location was displayed in the top left corner of the smartphone screen (see Figure 5.4, p.91).

In general, the amount and type of location information users are willing to share or would like to receive varies considerably. The suitability of collecting and providing such information depends, among other criteria, on the tasks the users are performing (Reilly et al., 2006). The presented evaluation focused on navigation throughout a small-size building and its immediate vicinity. In the next section, we describe a second user study that investigates and analyses the effects and details of run-time visualisation generation for users' chasing a moving object over a larger-scale mixed indoor-outdoor area.

5.4. Use case 2: Awareness of variability

The two goals of this real-life field trial were 1) to investigate the influence of awareness of the uncertainty about users' localisation on their performance and behaviour in a map-assisted *chasing task*, and 2) to evaluate the individual importance of the corresponding visualisation patterns, introduced in section 5.2.2, in providing this awareness.

5.4.1. Preparation and setup

Experimental environment and facilities

The study was conducted in a combined indoor-outdoor environment using a Samsung Google Nexus S Android smartphone running the application described in section 5.2. The indoor segment included a part of the university main building where a WiFi-based localisation was implemented as explained in section 4.4.2 and was configured to locate users every 4 seconds. The outdoor segment covered the campus area and was tracked by GPS that reported location approximately every 4 seconds. Within the context of the study, we assigned the walking speed V=1.2 m/s, which is around the lower end of the scale of the average walking speed (Patricia 2010). The timeouts for the transitions of the states of localisation visualisation in the case of missing updates are 5, 10, and 20 seconds, corresponding to 6, 12, and 24 metres of walking.

The participants' task (the *chasing* task) was to meet a dedicated person in the case of no direct communication available between the two, i.e. using only the information shown on the smartphone's screen. The dedicated person (hereafter referred to as the *target*) was also equipped with a Nexus S smartphone running the application.

Participants

Ten people (7m, 3f), aged between 24 and 60 (M=34, median=30.5), participated in the study. They were all employees of our university, specialised in diverse fields, and were not involved in our research. We asked the participants to rank their expertise in 1) using map-based applications for navigation, and 2) working with smartphones. The provided scores on a 5-point Likert-type scale – 1-none, 2-a bit, 3-some, 4-quite a bit, 5-a lot of – averaged to 2.9 (median=3) and 2.1 (median=1.5), respectively. The participants also ranked to what level they were familiar with the campus area and the building interiors (M=3.7, median=3.5).

Experimental conditions and procedure

The study began with a briefing, during which we explained the participants the goal of the study, introduced the chasing task and provided the main instructions on the application usage and functionality. The participants also met the target.

We compared two approaches to visualising location. In one, a user's last received location and its error are shown (Figure 5.2(a), p.87). The other visualisation extends the first one by providing additional feedback on the tracking status of each location as described in section 5.2.2 and illustrated in Figure 5.2(a-d). The main part of the study thus comprised two sessions, one for each visualisation approach. We will hereafter refer to the sessions as the *simple session* and the *extended session,* respectively to the visualisation used in each session. In the beginning of each session, we additionally explained the behaviour of the corresponding visualisation. The participants then had to walk to an indicated starting area outdoors, from where they began the chasing.

Wizard-of-Oz for manipulating the uncertainty Naturally, one's movements in a chasing task in a large-scale environment should be unrestricted as long as the corresponding areas are open to public. Therefore we let the participants plan their chasing strategy and navigation path individually in a way they felt most comfortable with.

Since location tracking was steadily available throughout the environment, we introduced artificial control over its availability so as to facilitate the tested conditions. Assigning a predefined static area may not work in the case of a freely chosen trajectory, for the latter might simply not run across that area. Therefore we simulated the absence of the participant's and the target's location updates using a wizard-of-oz (WOz) approach.

A WOz experiment is a type of experiment, in which a system's functionality or condition, which is supposed to work or happen autonomously, is in reality provided manually by an unseen human called the wizard-of-oz. In particular, this approach proved to perform well for location-based functionalities (Dearman, Hawkey, and Inkpen, 2005; Dearman, Inkpen, and Truong, 2010). Our WOz, played by the experimenter, manually chose a suspension area at runtime using the information about the locations of both parties. A candidate area for suspension had to satisfy one criterion: the target had to be roughly in the centre of the area at the moment of suspension. Figure 5.10 illustrates our WOz procedure.

Together with suspending the target's location updates, the WOz notified him about it with a vibrating alarm, and the target then proceeded to a previously agreed area (hereafter '*second area*') located out of sight from the current one. The target's location updates were brought back on the participant's smartphone as soon as the former left the suspension area and was on the way to the second area. It allowed the participant to see the target's locations again and thus to resume informed chasing. If the participant also entered the suspension area, the WOz suppressed their location updates as well while in the area.

We limited the areas where the suspension was possible to indoor locations in both sessions. This was done in order to avoid a direct line of sight between the participant and the target in an open area outdoors, i.e. when the participant could spot the target from a distance and thus ignore the application. Besides, Benford et al. (2006) found out previously that users of location-aware navigation applications tend to remember such problematic areas during their later experience in the same environment and may, for example, exploit these areas in the future for tactical advances or try to shun them. To avoid such effect of possible learning about the first session's "bad" area, a different and previously unvisited area was suspended during the second session.

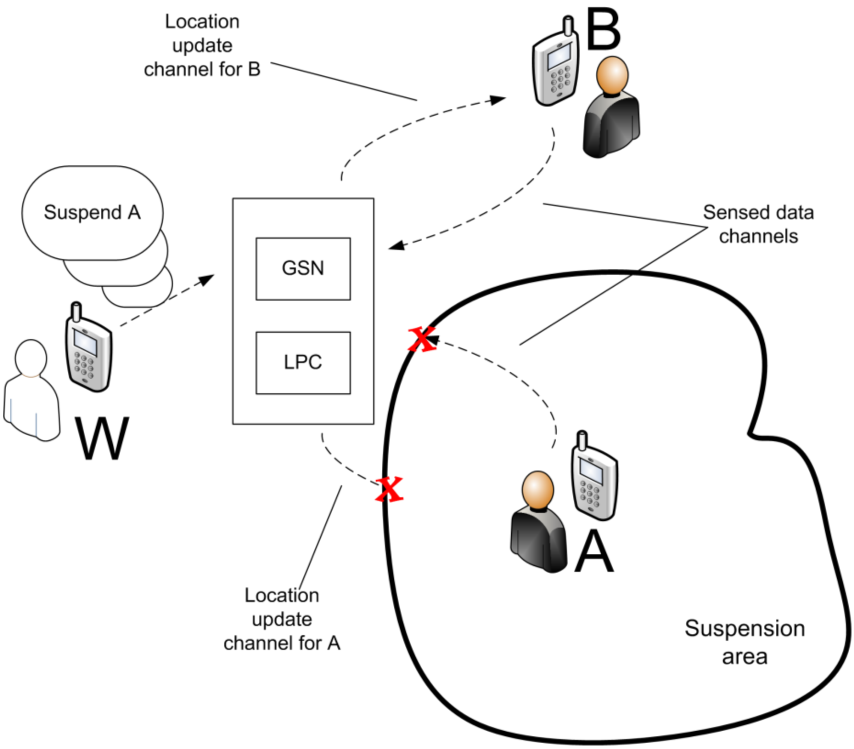


Figure 5.10. A wizard-of-oz (W) dynamically chose an area where users' location tracking would be suspended. Both the incoming and outgoing location data were blocked for or from the user who was in the area.

Each session completed when the participant met the target in the second area. On average, participants spent one hour to complete both sessions and answer all questions.

Thinking aloud and supervision Chasing a moving object in a large-scale and crowded area entails different path-choosing strategies. For example, the speed of walking, the experience of using mobile maps, the topology of the environment are among the many factors that may influence the chasing process. As we were mainly interested in analysing whether, how and in which situations visual awareness of uncertainty about localisation would be influential, we did not set any time constraints regarding the task completion to the participants. Instead, we instructed them to walk at a comfortable pace so that they would be able to pay enough attention to the information on the smartphone's screen. By doing so, we were also able to manage the "thinking aloud" (TA) approach during the sessions, in which the participants' oral comments were recorded and their behaviour observed for later analysis. We followed the speech-communication TA protocol (adopted from Olmsted-Hawala et al., 2010), in which the WOz acted as an active listener, replying a short and non-directive "um-hum" to the participant's TA comments. The same pattern, pronounced with a rising intonation, was used to remind the participant to keep TA after 15-20 seconds of silence. Walking next to the participant also allowed the WOz to observe the details of the participants' behaviour at run-time. Besides, due to the large-scale and spatially distributed nature of the experiment, the WOz was able to take appropriate decisions on the session flow in case of technical problems or difficulties (e.g., terminating the session if the participant was stuck and wished to withdraw or the application behaved inappropriately), but no other communication or interference into the evaluation process was allowed. Only two participants had prior TA experience so an appropriate explanation of the above conditions was provided.

Data collection

We collected four types of data:

Run-time information The run-time information comprised the "think aloud" comments and indirect observations and comments the WOz noted about the behaviour and experience of the participants during the sessions.

Post-session questionnaire The post-session questionnaire evaluated the participants' experience after each session and included the following questions to be ranked on a 5-point Likert scale (the final question had a 3-point scale):

- How difficult was this session to complete? (1-not at all, 5-very difficult);
- How helpful in achieving your goal was the information about the users' locations? (1-not at all, 5-very helpful);
- How distracting from achieving your goal was the information about the users' locations? (1-not at all, 5-very distracting);
- How confusing was the information about the users' locations? (1-not at all, 5-very confusing);
- How frustrating (annoying, stressing, discouraging, irritating) was the information about the users' locations? (1-not at all, 5-very frustrating (annoying, stressing, etc.));

- During the session, you referred to the information about the users' locations considerably (1-strongly disagree, 2-disagree, 3-neutral, 4-agree, 5-strongly agree);
- (if relative) How important was the fact that you were familiar with the environment? (1-not at all, 5-very important);
- (if relative) Would you use this type of users' run-time locations in a similar task in an unfamiliar environment? (Yes, No, Difficult to say).

Post-experiment evaluation block The post-experiment evaluation block measured the usefulness of the visualisation patterns in the extended session in terms of how they assisted in the chasing task during this session. We asked the participants to rank how much they agreed with the following statements below, using a 5-point Likert scale (1-strongly disagree, 2-disagree, 3-neutral, 4-agree, 5-strongly agree):

(1) The dot was helpful;
(2) The extending circle was helpful;
(3) The cross was helpful.

Since the performance in a large-scale environment may be influenced by a lot of factors indeed, we also asked the participants to support each score with a short explanation. This supplementary information was necessary in order to reveal whether the given score was influenced by external factors (i.e. unrelated to the awareness, such as being dissatisfied with poor tracking, hurdles in the facilities, etc.) which, if not found out, could have led to a misinterpretation of the results.

Post-session and post-experiment comments in free form Finally, the participants could also provide any additional comments and suggestions they had.

5.4.2. Results and observations

Findings summary

We derived the following main findings from the study:

- The extended design with the visualisation patterns, introduced in section 5.2.2, p.86, is equally easy to use as the basic circle-based one (see Figure 5.1(i), p.83), and additional awareness of the uncertainty about location is beneficial;
- Different constituents of this awareness are of different importance; therefore the corresponding visualisation patterns should be given different priorities when designing a user interface that contains uncertainty about location;

- The importance depends on user profiles (such as their eyesight level, navigation skills, ability to understand maps, etc), distance between the users, and the quality and reliability of the tracking.

We will now describe and discuss the results and the above findings in detail.

Overall performance

Eight participants successfully completed both sessions. One participant (P8) could not complete the simple session due to an issue with the network that led to the application on the target's smartphone being unable to resume sending location updates after the target had left the suspension area. This, however, influenced P8's experience during the extended session and brought in several valuable observations for later analysis. One participant (P7) could not complete both sessions: the target could not follow the instructions due to the rain that intensified after each session had already started, so he did not walk to the agreed second meeting area outdoors. The WOz was unaware of the target's incomplete path and therefore made a mistake in the suspend-resume process. This resulted in highly incorrect positioning so that the participant could not find the target and gave up. We had to discard P7's results because he rather evaluated the impact of erroneous information.

Figure 5.11(a-g) compares the scores provided to the per-session questions. The Wilcoxon signed-rank test applied to the scores didn't reveal any difference between the two designs, what suggests that the extended design comprising the additional three visualisation patterns is at least as good to use as the basic one (see Figure 5.1(i), p.83). This allows us to consider and analyse the added value the visualisation patterns bring into the extended visualisation.

All participants referred to the additional awareness in all occasions where they recognised and experienced it during the extended session. The results in Figure 5.11(1-3) show that regardless of the experience during the tasks, all three involved visualisation patterns were considered helpful with respect to the information they intended to provide:

- **Dot (mean=3.7, median=4, σ=1.41).** Three participants, when explaining the corresponding score, commented that they could not see the dot or just did not pay attention to it so they simply did not notice it at all (hence a higher σ value). The rest (strongly) agreed that it was useful. *"I like the heart bit of my location now." – P6, "Status indication of signal assured correctness of location." – P8*. In particular, because of the negative experience during the simple session, P8 indicated that he was especially concerned that the dot be on, *"[I was] afraid of missing signal indication reliability."*

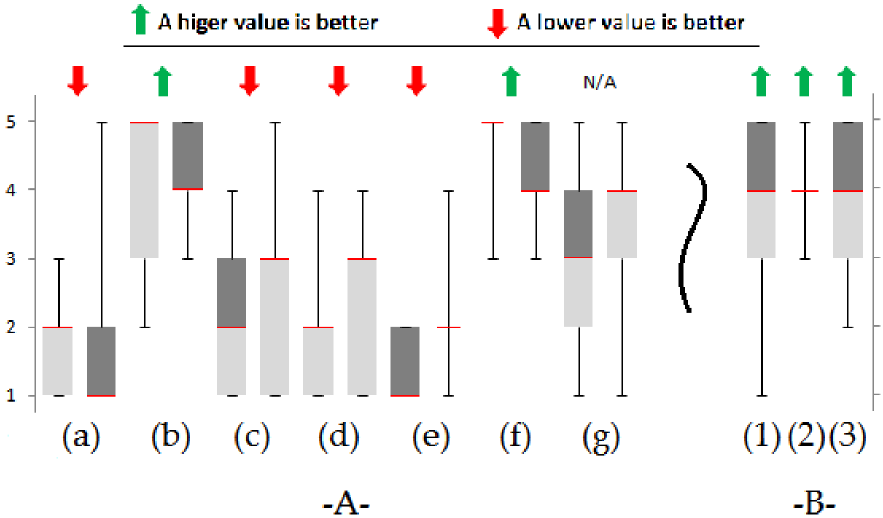


Figure 5.11. A: Comparison of the 1-5 Likert scale scores to the questions after each session (left – simple session, right – extended session): (a) – session difficulty, (b) – helpfulness of location information, (c) – distraction level, (d) – confusion level, (e) – frustration level, (f) – reference to location information, (g) – familiarity with the test environment; B: The 1-5 Likert scale scores for the involved visualisation patterns in the extended session: (1) – dot helpful, (2) – extending cross helpful, (3) – cross helpful.

- **Circle (mean=4, median=4, σ=0.7).** Two participants expressed that they did not refer to the extending nature of the circle much and therefore could not conclude that that property assisted them. However, they did explore the area of the map covered by the circle. For example, P5 realised that the target, while being shown indoors, was actually outdoors, i.e. near the circle's edge (see Figure 5.12). The rest regarded the dynamically extending circle as a useful indication of the location tracking dynamics. *"The first [extended] session was easier to use because of some continuous changes." – P2.*
- **Cross (mean=3.9, median=4, σ=1.05).** Overall, the cross served the intended means of awareness – to communicate the information about an outdated (missing) location – as in *"I can see target not there, so change strategy to locate target" – P4,* or *"Unavailability of signal for a longer time indicates that we cannot rely on the location anymore" – P8.* P5's negative score (2/5) was brought by her confusion about the time when she stopped receiving updates from the target, *"It confused me that he [the target] disappeared, so how can it [the cross] be helpful?!" – P5.* We clarify on this response further on below when we talk about the cross' importance.

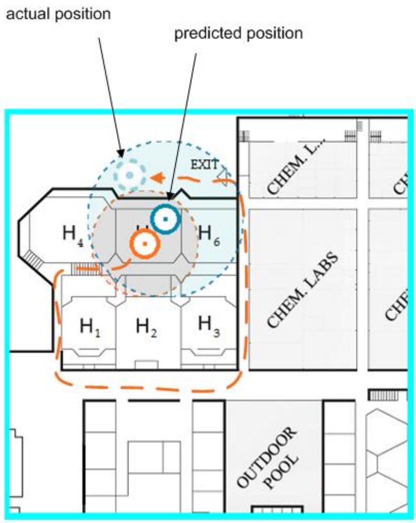


Figure 5.12. One of the participants realised that the target, while being shown indoors, was actually outdoors, i.e. near the circle's edge, so she left the auditorium and proceeded towards the nearest exit, following the map.

Individual evaluation

Each visualisation pattern turned out to be of different importance. The dot indicating that the location update was recent was ignored by the participants who could not focus or felt uncomfortable focusing on that detail. Also, it was mainly invisible outdoors in the sunlight. Although the participants did not reflect this fact in the ranking, they did mention it in the beginning of a sunny session. Moreover, since the participants started outdoors and the target remained indoors in the beginning of the session, then, prior to entering the building, they often carried the device in their lowered arm and only occasionally raised the hand up to look at the screen. Thus the importance of the dot (given that it can be perceived if shown) increases in problematic areas where location tracking may become unreliable or absent, or in situations when participants are close to each other and consult the map more often.

In a similar way, participants paid less attention to the circle's gradual extension while location tracking was stable. However, its importance also increased in the problematic areas, what can be illustrated by an observation linked to the target's suspended period in the simple session. In that case, while no updates were made to the visualisation of the target's position, his resumed updates outside of the suspended area surprised the participants, for the new position appeared at quite a distance from the last known (i.e. prior to the suspension) location – *"What was it? Did he move so fast or what?" – P3, "He moved!" (a remark during the session),"It was a bit annoying that in one second the object was in one position and just after it [he] moved without visible movement" (a comment to the ranking) – P9, "Predicted spot was flipping – sudden location change" – P10.* And the extending circle gradually transforming and replaced, after a time, with a cross gave the feeling of movement and dynamics and created a firm positive attitude. In other words, it prepared the participants for the change, smoothing the cause of no location tracking.

In the case of lost tracking, i.e. when the circle eventually turned into a cross, the participants reacted differently. For example, P4 tried to estimate where he should look now and did not go to where the target's cross was shown. P6 admitted that she was just guessing. P9 did nothing, stood still and waited for the target's location to resume. Somewhat apart here stays P5's reaction, who disagreed that the cross was helpful. But when we asked her to explain the given score, it became apparent that it was the very fact that the target could not be tracked that made her unhappy (see the quote above), but the awareness about it was recognised and accepted (which she did notice as can be seen from her comment about that moment, *"How come is he here? Hmm...Oops, the cross now. Did he disappear?!"*). Besides, she agreed that seeing the blinking dot indicating that the location tracking was active was also very helpful (5-strongly agree). Therefore, we conclude that the cross for the case of missing tracking updates was important for the awareness and should always be activated.

Other findings

None of the participants refused to try the task in a completely unfamiliar environment, with 7 out of 9 agreeing to use the extended design, of which 5 would also try the simple one. Moreover, P10 supported his "Difficult to say" choice by saying that he never needed that kind of information in his activities, but nevertheless he expressed an interest in repeating the test in an environment unknown to both parties. Similarly, P4 concluded that knowing the environment was not very important to him with respect to completing the task (*"I think environment is not important factor; using this [extended] scenario, target's changing location can be seen on map, and more info than before [simple]"*); therefore he was also curious to validate his scores in an unfamiliar environment.

There were situations, in which additional awareness, though recognised, was rather bewildering. For example, P8 in his failed simple session got perplexed by not having found the target even within the precision area. P2 admitted that seeing the additional awareness was quite helpful as it allowed knowing more about the movements. But at the same time he explicitly mentioned that the perplexity in the extended session was caused by imprecise GPS positioning near the entrance (he stood quite close to the wall), so he preferred the simple mode because of more accurate location detection. Such observations of reacting to the application's inappropriate behaviour, in fact, are related to the problem of intelligibility in context-aware applications (e.g., "Why is my GPS position so far from where I actually am?" – "You are standing too close to the building"), where Lim and Dey (2011) showed recently that providing intelligibility increased users' impressions about a context-aware application with low certainty when it behaved inappropriately.

5.4.3. Conclusions and discussion

The presented field study evaluated our approach to incorporating *awareness of uncertainty* about location into mobile graphical user interfaces. Assisted by a smartphone running a map-based application, the participants completed two tasks under varying location tracking conditions in a mixed indoor-outdoor environment. In each task, the uncertainty was visualised according to either the referred basic technique (see Figure 5.1(i), p.83), which previously proved to be beneficial for navigation tasks (Dearman et al., 2007), or the experimental extended design introduced in section 5.2.2. The user interface adaptations to presenting the varying levels of awareness and uncertainty in the extended design are accomplished by mapping the capabilities of localisation systems and location data onto an ontology. The ontology is used to produce a set of user interface elements, called *visualisation patterns*, which represent the characteristics of uncertainty and can be used as part of a complete user interface.

The results of the study show that the extended design turns out to be at least as good to use as the basic one, i.e. the additional visual information is not considered a burden. The evaluation of each visualisation pattern reveals that their impact depends on each user's personal profile (such as their eyesight level, navigation skills, ability to understand maps, etc.), distance between the users (knowing about finer details of someone else's state is less important if they are far), and the quality and reliability of the tracking. On the basis of these conclusions and observations, we identify a set of guidelines for presenting the uncertainty of location in a user interface.

We have so far manipulated with the variability of location context alone. However, location information needs can be affected, sometimes considerably, by various details, often considered in combination with each other (Reilly et al. 2006). For example, using information about a user's daily agenda, such as Lovett et al.'s (2010) shared online calendar, or location-based activities (e.g., Dearman and Truong 2010, Dearman et al. 2011), one may learn that the addressee must not be disturbed. A privacy setting obtained from this user's privacy preferences may request to change the visualisation mode (Tang et al. 2011) or even make location information within a certain area intentionally unavailable at all times (Brush et al. 2010). The degree of correctness or importance of particular location information depends on this information owner's perception of it. In the next chapter we show how the variability of location context can be successfully combined with users' social details in offering context-aware information sharing in a run-time help scenario in the area of vehicular ad-hoc networks.

Chapter 6

Geo-social interaction: combining location and social context

A shorter version of this chapter has been published in the following conference proceedings:

- Mahmud, N., Aksenov, P., Yasar, A., Preuveneers, D., Luyten, K., Coninx, K., and Berbers, Y., 2010. Geo-social interaction: context-aware help in large scale public spaces. In *Proceedings of the 1st International Joint Conference on Ambient Intelligence,* pages 107–116.

This chapter describes our joint work completed in collaboration with our colleagues from the DistriNet research group at KULeuven, Belgium. We at EDM defined information needs and requirements for employing the knowledge about the locations and movements of vehicles and about their drivers' social profiles within a ubiquitous help system in order to determine relevance of the respondents to a help request. The colleagues from DistriNet identified the requirements of vehicular networks, realised message routing in such a network taking into account both contextual information and network message passing techniques, and validated the proposed approach in a network simulator.

Together, we showed how combining location and social context can improve information filtering in a large-scale vehicular ad-hoc network.

6.1. Problem statement

Being able to obtain the right information at the right time has always been a challenge when taking informed decisions. Unfortunately, people who are on-the-move often do not have an opportunity to spend a long time on search. Moreover, efficient delivery of timely and relevant information may be also used in various optimisation scenarios, e.g. to optimise traffic flows. The proliferation of wireless networks has created new opportunities for complex peer-to-peer information dissemination systems, and a key challenge in this area is how to interact, locate and communicate effectively in a large-scale public environment.

As discussed in section 1.1, context-awareness is the capability of a system to decide what information is relevant and to use this information to provide relevant services and correct solutions. The recent huge success of socially oriented applications and services, such as social networks, has put the social context in the limelight of efforts on context-awareness. A social network is a map of so called social relationships among members of a network, which make invisible interpersonal relationships visible to the real world. Intertwined with information on people's locomotion, social context-awareness has led to the concept of location-based social networks and services, where interaction, collaboration and information sharing are determined by people's locations. Examples of research in this area are the Connected Traveler project (Manasseh, Ahem, and Sengupta, 2009), a trip planning tool that provides relevant information to drivers (e.g., about traffic congestion) and public transport users (e.g., about expected bus arrivals) by taking their quickly changing location and personalised preferences into account. Connecto by Barkhuus et al. (2008) is a phone based status and location sharing application, which evolved, throughout use within a small group of friends, from serving for mere location updates into a tool for enriched social interaction through these location-based updates. CityFlocks by Bilandzic, Foth, and De Luca (2008) is a mobile system that allows information seeking visitors to access tacit knowledge from local people about their new community. Commercial applications, such as Foursquare[26] and Gowalla[27], are based on so called check-ins, when mobile users share their current location (at a thus checked-in place) with friends and may also leave their opinions about the places they have checked-in, which proved to be a very attractive service (e.g., as of June 2011, Foursquare had 10 million registered users with an average of 3 million check-ins per day).

These and many other examples show that the area of using the location and the social contexts together is very broad. In this chapter, we investigate how the knowledge about users' locations and movements and the information about their social profiles and personal preferences can be combined and employed to help users in a large-scale highly dynamic public space obtain timely and relevant information. In particular, we determine the most appropriate individual as a possible assistant in a 'help me!'-like scenario between members of a vehicular network. We analyse peers' spatial proximity and explore their social connections in order to find an individual who specialises in the same or the closest

[26] http://foursquare.com

[27] http://gowalla.com

area as specified in the help request and who has trusted reputation; we then take into account information about their locomotion, such as their current location and direction and the required travelling time.

The next section presents a close-to-life scenario of a situation where such help may be required, and in the subsequent sections we describe our approach in detail and then report about the results of a simulation that we ran to validate the approach.

6.2. Motivating scenario

Raymond is a family doctor. To visit his patients, he drives regularly to many places throughout the city and in the country-side. Today, he received a request from Mr. Johnson whose daughter became ill. While Raymond was driving through the city centre, his car broke down. Luckily, he was able to identify the cause of the breakdown and apparently he knows how to fix it, but unfortunately, he discovered that the required tool, specific to his car, is missing from his repair kit. Since Raymond needs to continue the trip as soon as possible, he does not have enough time to wait for the official repair service to arrive. He cannot take a taxi either, for he will have to go to the next appointment later. Instead, Raymond decides to use the Ubiquitous-Help-System (UHS), a dedicated social network of car owners, and to try to find someone who happens to be driving in the vicinity and therefore may be able to lend him the tool faster. He briefly describes the problem and the tool he needs and sends out a help request, which he sets to be valid for the next 10 minutes. The UHS application passes the request on to the people listed as Raymond's friends in the network. Since the network of the UHS users is interest-based, his friends there drive the same or a similar car model. However, though even available, they might be far enough from the place and simply unable come. Depending on the similarity of the car models and their current location and availability, each friend's UHS computes how likely it is that this particular friend can come and provide the necessary help to Raymond and sends back the result. Additionally, each friend's UHS sends Raymond's request further to their friends and the responses also arrive to Raymond's UHS client. The application then selects the best match and displays the details to Raymond, showing the car type and the expected time of arrival. Raymond confirms the selection, waits for around 8 minutes for the driver to arrive, they fix Raymond’s car and he continues his trip.

6.3. Aspects of geo-social interaction

This section discusses a number of aspects that we have to consider in creating a context-aware help system suitable in large-scale public spaces.

Spatial coverage It is always desirable to know an accident's exact location. For instance, in the case of an accident on the road, the authorities should be notified about its exact location so as to react as soon as possible, and the information should be delivered only at the right place. Similarly, a context-aware application should be able to sense, manipulate and disseminate the information about vehicles' directions and cruising speeds in order to be able to prevent traffic congestions or accidents in specific regions.

Timeliness It is crucial that the information being disseminated in a large-scale network between (groups of) nodes reaches the destination in time. Timeliness uses time as a relevance criterion for information sharing in order to ensure that the information is not older than the 'lifetime of the information', i.e. to guarantee that the right information is delivered at the right time.

Completeness A lack of information can lead to ambiguity. Completeness verifies the quality of information provided by a node in the network and, for example, may compare the number of attributes received to the total number of attributes required to make an informed decision.

Trust-worthiness The information being received at a particular node should be reliable and trustworthy. Each node in the network must have a social profile describing their interests with a certain quality of information.

Significance Significance indicates the importance of the particular contextual information required by a node in the network. The significance is increased in the case of a life threatening situation.

Making Friends A social network allows its users to find friends of a friend, too. This feature makes it possible to find people with similar interests and allows one to make new friends interactively.

Grading Friends Information recommended by a friend is naturally perceived to be more trusted. In a social network, peers also produce and share information. A peer producing information is also subject to trust. A grading system, which depends on the feedback voluntarily provided by the information consumer, helps to improve the overall quality of information shared in the social network.

Distributed Feature The friend-of-a-friend (FOAF) concept allows the information about a friend in the network to be distributed without a centralised database. In a dynamic large-scale pervasive computing environment, where the number and the availability of nodes of the network at a given time are unpredictable, the distributed feature lets a UHS peer find and interact with other peers with similar interests in a manner close to how friends in a basic social network are approached.

6.4. The three-leaved mirror approach

We present an approach, which we called the three-leaved mirror approach, to tying the spatial and the social contexts so that each resembles one of the two side leaves of such a mirror.

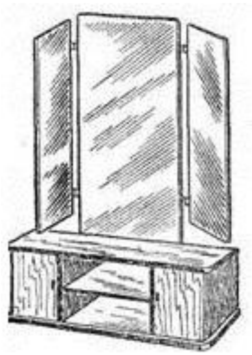


Figure 6.1. Each of the side leaves of a three-leaved mirror can be adjusted independently in order to change which exactly part of the user is reflected in the leaf's mirror.

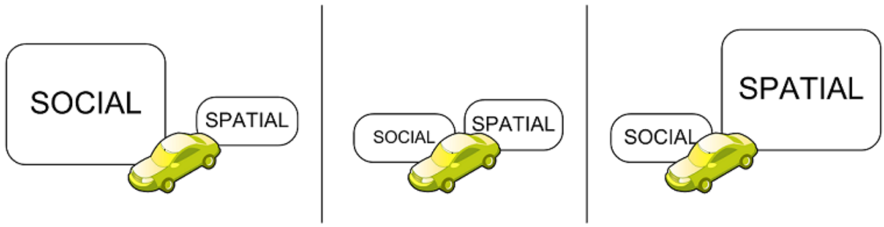


Figure 6.2. A peer's spatial and social contexts contribute differently to this peer's relevance as a potential help provider.

Like it is with a real three-leaved mirror, whose angles can be adjusted independently to change a user's current view on each side (see Figure 6.1), we vary the contribution of peers' spatial and social context to the overall relevance of their assistance (see Figure 6.2). And the central leaf then optimises the information flow in the network by reducing irrelevant information dissemination.

6.4.1. The side leaves: geo-social approach

Like network availability, location-awareness has nowadays become a ubiquitous property of many mobile devices. They come equipped with both networking and location detection technologies, so that we can easily access the information available on the user's location and the details specified in the social network. For example, we can select a contact that would be in the vicinity and then ask this person for assistance. The difficulty lies in dealing with the additional complexity caused by people’s permanent movements and by the timeliness of certain requests for assistance (e.g., if no help arrives within 5 minutes, the requester will have to manage on their own). We used vehicular networks for evaluating our approach. Since this type of network is highly dynamic, contains permanently moving objects and requires a close eye on the timeliness, it is a suitable evaluation framework.

6.4.2. The central leaf: improved relevance backpropagation with geo-social relevance

Geo-social interaction requires communication between people, therefore optimised information dissemination plays an important role. There are several information routing strategies, which take, or do not take, the quality of information (QoI) into account, such as broadcasting and backpropagation, respectively. The QoI-based best-effort mechanism disseminates context using a relevance function (introduced later in section 6.5). Each participant has a list of friends (e.g., in their Friend-of-a-friend, or FOAF, profile), a score value and properties about the relevant context information they can provide (see Figure 6.3). The information is forwarded to the adjacent nodes who are either friends or friends-of-a-friend having a certain score value unless a maximum number of hops is reached. Each forwarding node reduces the hop counter, adds its identifier and marks the message relevancy tag if the information is relevant for its purpose and grades the sending node positively, adding it to the friends list. The feedback technique is based on the spatiotemporal and social context information, such as position, speed, direction, time-to-live, and interest, that are combined together to decide whether the received data would be relevant. It also helps to determine information relevancy on the intermediate nodes. The feedback to the delivering node is initiated if the context information is relevant, irrelevant, unused, or duplicate information is received. It ensures that the provided information is from a trusted node and is supposed to be accurate and relevant for the receiver.

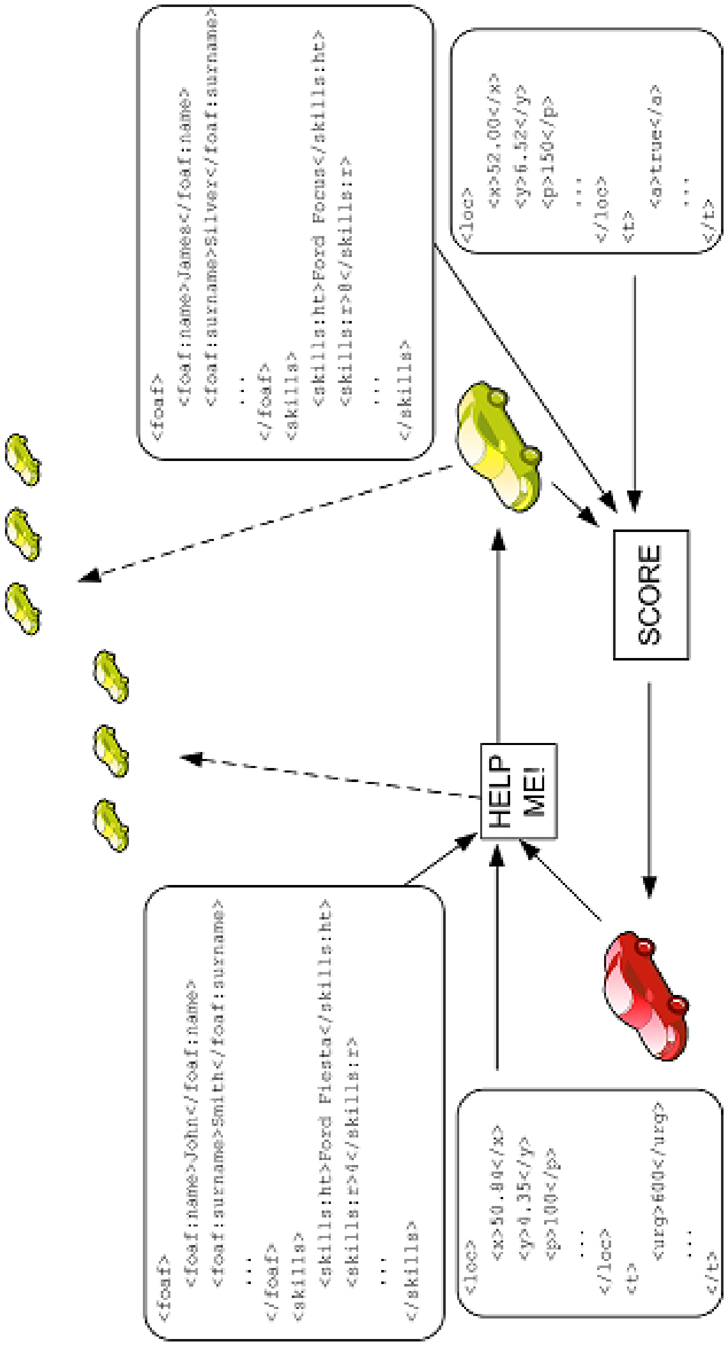


Figure 6.3. A help requester sends their location and the details of their social profile to their friends. Friends return a relevance score calculated on the basis of their availability, social similarity and spatial proximity. Friends also send the help request to their own friend networks, using the FOAF profiles.

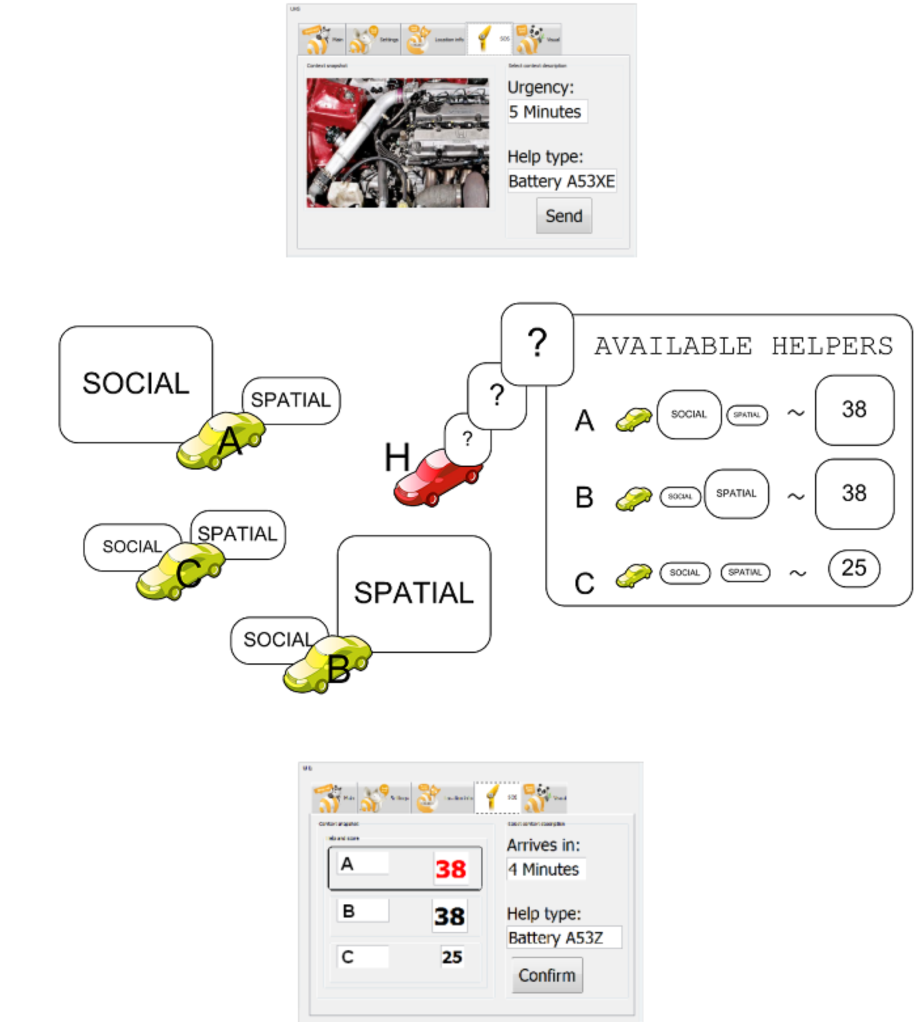


Figure 6.4. Talking cars. (a) Help seeker H sends a help request using an embedded Geo-social UHS; (b) A, B, and C are ready to help H; (c) Depending on the returned geo-social relevance score of the helpers, H confirms the offer.

6.4.3. The ubiquitous help system

The ubiquitous help system (UHS) with geo-social relevance, which realises the proposed variation of social and spatial contexts, allows drivers to receive positive help responses, according to how close or how knowledgeable potential help providers are. After sending a help request, the help requester receives a list of users who are able to help, from which they choose the one that suits best. Figure 6.4 shows examples of using the UHS client for creating a help request, of the client's intermediate process of ranking received responses, and of choosing the fittest assistant.

6.5. The geo-social relevance function

The key concept in our approach is the geo-social relevance function that defines how relevant a potential help provider is to the help seeker. The function naturally links together several parameters from each of the three leaves (spatial, social, and network-bound) and calculates a relevance score that each involved node of the network has with respect to the help requester. The higher the score, the better the provider. The function is expressed using the following formula:

$$GSR(peer) = A \cdot \sqrt[n]{R \cdot HT} \cdot F_U$$

The meaning of the multipliers and the root index, *n*, in the right part of this formula are explained below.

Availability (A)

Availability of a node is a Boolean value that simply indicates whether the corresponding node can be a potential help provider. We assume that if a peer is unavailable (A = 0), the help request is still passed further to this peer's friend-list.

Reliability (R)

Reliability of a node in the network is a peer-determined integer value between 1 and 10 indicating how helpful the corresponding node has been in the past.

Help-type (HT)

The help type value measures the requester's and the provider's technical match and is an integer between 0 ('I know nothing') and 10 ('A perfect match').

Root index (n)

Index n stands for the number of hops in the network between the requester and provider. The reason for choosing the root-based value for measuring the contribution of the social parameters to the overall relevance is that the level of trust to somebody who is connected to you indirectly decreases significantly.

Spatiotemporal contribution (F_U)

The contribution of location information is defined as

$$F_U = \begin{cases} 0 & \text{if } Direction = 0 \\ 0 & \text{if } U \cdot V \leq D_{min} \\ e^C & \text{if } D_{min} \leq U \cdot V \leq D_{max} \\ e \cdot C & \text{if } U \cdot V > D_{max} \end{cases}, \text{ where } C = \frac{U \cdot V - D_{min}}{D_{max} - D_{min}}$$

Here, ***Direction*** equals 1 if the provider is moving in the direction of the requester, and 0 otherwise. ***Urgency U*** is the time interval within which the help is needed; its value is specified by the requester. ***Velocity V*** is an estimated average velocity of the help provider during period U. We assume that the help requester does not move (e.g., their car broke down). The corresponding maximal and minimal distances, D_{max} and D_{min}, between two nodes are calculated at the moment when the help request has been received; they depend on the spatial topology of the area, such as, for example, the actual length of the connecting path between the nodes that might be affected by possible repair works, closed or blocked roads, etc.

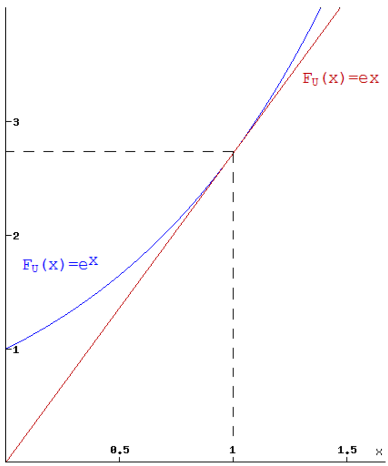


Figure 6.5. $e^x > e{\cdot}x$ for x € [0; 1) and thus e^x has a bigger weight for the score; and for $x \geq 1$, the smaller value of $e{\cdot}x$ is used.

The logic behind the expression for F_U is that $e^C > e{\cdot}C$ for nodes that are far (see Figure 6.5), i.e. for which $0 \leq C < 1$, and thus e^C has a bigger weight for the score; and the smaller value of $e{\cdot}C$ is used for $C \geq 1$, thus making the social parameters weigh more than the spatial ones when comparing the scores of two different nodes. Besides, the score function makes all members of the network – the requestor (urgency U), the potential providers (availability A), and all other nodes in the network (reliability R) – collaborate implicitly in finding the fittest solution to the help request.

6.6. Validation

Testing large-scale groups in real-life conditions is not easy and not always possible. We evaluated the performance of our geo-social relevance approach using a real-time discrete event-based network simulator (OMNeT++)[28] that ran on a large-scale vehicular network using a realistic dataset (Raney et al., 2002) logged for a period of 24 hours. The spatial data in this dataset are distributed over an area of 250 by 260 km. All individuals choose a time to travel and a route according to their place of residence and the current road congestion. The complete dataset contains more than 25 million records of locomotion of 260.000 vehicles, from which we randomly selected 300 vehicles. Also, for simplification purposes, we scaled the selected data from the original 250x260 km to fit [0; 1].

6.6.1. Data preparation and setup

This section describes the details of the actual data we used in the simulation, on the basis of the requirements listed in section 6.3 and the details of the geo-social relevance function in section 6.5.

Spatial arrangements

The provided GPS readings of nodes' movements contained only timestamped 2D-data. On the basis of these readings, we derived other parameters required for modelling further variability of location context. Thus, the velocity of a node at each recorded location was averaged by this node's two subsequent movements (see Table 6.1). It gave an acceptable approximation of the time help requests remained valid for.

The original data did not report anything on the location error of the provided measurements. Therefore we divided the entire area into three sub-areas so that in each area a measurement's error belonged to a specified interval. The intervals were assigned values from (0.0001; 0.0005), (0.0005; 0.001), or (0.001-0.005), respectively (see Figure 6.6), where 0.0001 equals approximately 25 metres. The error stands for the radius of a circle centred at the detected location, so that the node's actual location is within this circle.

[28] http://www.omnetpp.org

Table 6.1. Average velocity calculation

time	x	y	v_{avg}
15443.00	0.511	0.883	0.0000950
15450.44	0.510	0.882	0.0003450
15472.00	0.510	0.882	0.0003450
15474.05	0.509	0.881	0.0001280
15481.62	0.509	0.881	0.0000952
...	...	...	...

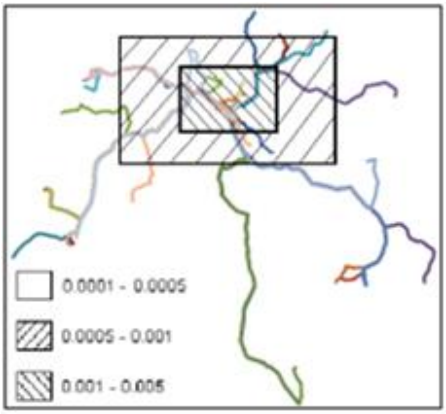


Figure 6.6. Location errors, with ranges 0.0001-0.0005 (outer area), 0.0005-0.001 (middle area), and 0.001-0.005 (inner area) where 0.0001 equals to approximately 25 metres on the original scale.

Since we did not have a description of the spatial topology of the area, we used a simple Euclidian distance (see Figure 6.7). Together with the assigned measurement errors, it gives the following formulae for computing the D_{min} and D_{max} values between two nodes at the time of calculation:

$$D_{min} = min(0, D_{mes} - P_A - P_B)$$
$$D_{max} = D_{mes} + P_A + P_B$$
$$D_{mes} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

where P_A and P_B are the errors of the corresponding measured locations of the nodes in question.

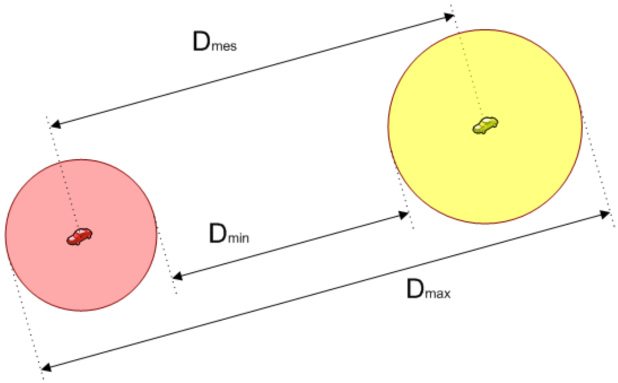


Figure 6.7. Distances between two nodes are simple Euclidian distances. In the case when the precision circles overlap, Dmin is 0.

Help-type

We introduced nine help-types so that each of the 300 nodes belongs to one of them. The total number of nodes was distributed normally among each help-type; Table 6.2 shows the distribution.

Table 6.2. Distribution of nodes into help-types

Help-type	1	2	3	4	5	6	7	8	9
Number of nodes	11	26	43	55	55	45	32	21	12

The matching table of the HT values corresponding to a pair of help-types is shown in Table 6.3. Notice the asymmetric nature of the help-type values for the corresponding (ReqHT to PrHT) and (ReqHT to PrHT) pairs in Table 6.3; the assumption behind the asymmetry is, 'If you are able to help me with my problem, it does not guarantee that I am able to help you with yours.'

Friend-list

In general, the number of friends in the friend-list of a node in the network is reasonably not limited. However, taking into account the specifics of our simulation, which was performed on a desktop computer with 2.4 GHz dual-core processor and 2 GB of memory, we had to limit the capacity of each node's friend-list to 15 friends, so that the simulation runs smoothly. The initial number of friends was assigned based on the type of help a node can provide. Friend-lists are extended in an asymmetric way: each time a help requester has received help from an appropriate provider, the latter is added to the requester's friend list. In the case of a new help request from the same node, chances are that the same provider can help again, and contacting this provider directly will save time and resources. Table 6.4 shows the initial friend lists of the first five nodes.

Reliability

An integer value between 1 and 10 was assigned randomly to each node's reliability characteristics. In general, a node's reliability increases or decreases dynamically based on their performance as a helper, but for the purposes of our simulation, we kept the values static throughout the simulation.

Urgency

Out of the 300 nodes, we picked 10 which would, each at a random point during the simulation time, become a help requester with an individual urgency of their help request. The urgencies of 600, 300, 120, 180, 120, 120, 120, 90, 180, 540, and 240 seconds, were used.

6.6.2. Details of simulation runs

The nodes in the simulator move around like cars and establish and close connections in accordance with their range to other nodes. The parameters considered for each node are: (i) time, (ii) velocity, (iii) (x;y) coordinates, (iv) number of messages sent, (v) number of messages received, (vi) number of forwarded messages, and (vii) time-to-live (urgency). Some nodes act as context providers and some as context receivers. All nodes forward the information to their peers as long as the urgency, given that all other context constraints are met.

We ran the simulation in three conditions: (1) with simple relevance back-propagation, (2) with the relevance back propagation extended with the experimental geo-social relevance ranking, and (3) with the state-of-the-art baseline case of plain broadcasting. Each condition evaluated a period of 24 hours of the vehicles' movements. Only relevant context information is disseminated in the propagation algorithms.

There are several types of messages in the network: *sent* (M_s), *unique received* (M_{ur}), *unique sent* (M_{us}), *forwarded* (M_f), *duplicate* (M_d), and *dropped* (M_{drop}). We measured a set of major network metrics: *Availability (A), Network Traffic (NT), Message Distance (MD), Trustworthiness (T),* and *Relevancy (R),* expressed in terms of the network messages:

$$A = {\sum_n M_{us}} \Big/ {\sum_n M_{ur}}$$

$$NT = \sum_{n}(M_s + M_f)$$

$$MD = {\sum_t Edges}\big/{\sum_t Nodes}$$

$$T = 1 - {M_f}\big/{\sum_f(n)(M_{ur} + M_d)}$$

$$R = {\sum_n\left((M_{ur} + M_d) - M_{drop}\right)}\big/{\sum_n(M_{ur} + M_d)}$$

Figure 6.8 illustrates a simplified visualisation of the simulation in action. The red cars are the ones broken down who need assistance. The yellow cars are possible help providers.

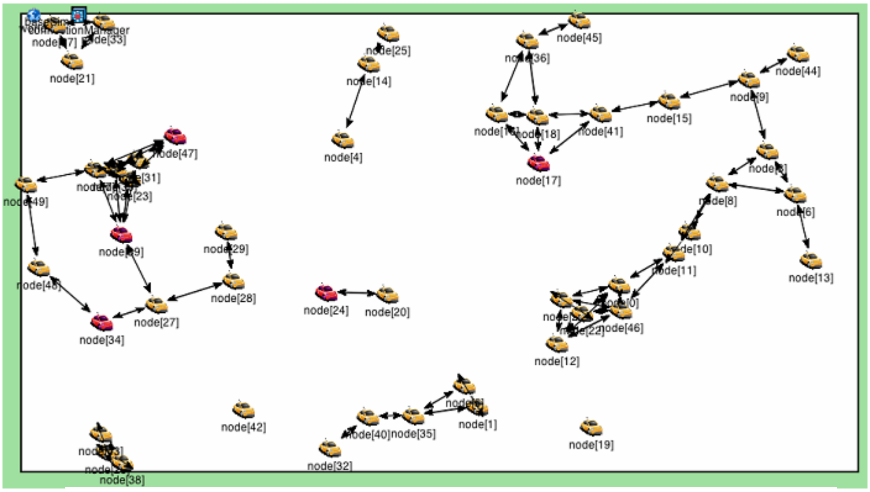


Figure 6.8. Simulated experimentation using OMNET++.

Table 6.3. Pair matching of help-types.

	PrHT1	PrHT2	PrHT3	PrHT4	PrHT5	PrHT6	PrHT7	PrHT8	PrHT9
ReqHT1	10	0	8	8	8	0	0	7	0
ReqHT2	8	10	0	0	0	8	6	8	4
ReqHT3	8	0	10	9	7	0	8	6	8
ReqHT4	9	4	9	10	9	0	4	0	8
ReqHT5	6	6	4	7	10	6	9	8	0
ReqHT6	7	8	0	4	0	10	7	9	9
ReqHT7	4	7	7	8	6	8	10	0	6
ReqHT8	0	9	0	0	4	9	8	10	7
ReqHT9	0	0	6	6	8	7	0	4	10

Table 6.4. Initial distribution of friends into friend-lists for the first five nodes.

Node	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
1	79	117	93	0	0	0	0	0	0	0	0	0	0	0	0
2	10	1	54	74	126	165	184	0	0	0	0	0	0	0	0
3	10	4	77	44	88	84	189	177	0	0	0	0	0	0	0
4	11	64	186	0	0	0	0	0	0	0	0	0	0	0	0
5	74	76	113	97	150	273	0	0	0	0	0	0	0	0	0

6.7. Results

We achieved a considerable improvement for each metric's performance using the experimental improved relevance backpropagation, compared to both simple relevance backpropagation and plain broadcasting. For example, the utilisation of the Network Traffic is 90% of that using simple relevance backpropagation and as low as 50% of that of simple broadcasting. The five left plots in Figure 6.9 visualise such comparisons for all five metrics on the percentage scale.

We also measured the behaviour of the geo-social relevance. The geo-social relevance scores were calculated and stored in all three conditions of the simulation, but were involved only in the experimental improved back propagation. Some returned scores were quite high in simple relevance backpropagation algorithm (e.g. 96) and low in the improved relevance backpropagation scheme (e.g. 16). This meant that in some cases unexpected or unplanned nodes, which happened to be close enough, had a better match than the algorithmically chosen ones, but averaged over all returned values, the improved algorithm outperforms the other (see the right plot in Figure 6.9).

6.8. Discussion

The results show that the improved information dissemination approach taking into account drivers' both location and social context achieves a considerable improvement in several main aspects of quality-of-information (QoI), such as relevancy of the provided information, the network distance a message in the network has to travel, the network traffic, availability, and trustworthiness. By eliminating redundant and irrelevant information sources in terms of their spatial and social similarity/difference we can limit information dissemination to within a much smaller number of nodes, which additionally are reliable and trustworthy, thus improving the overall performance of a large-scale communication network.

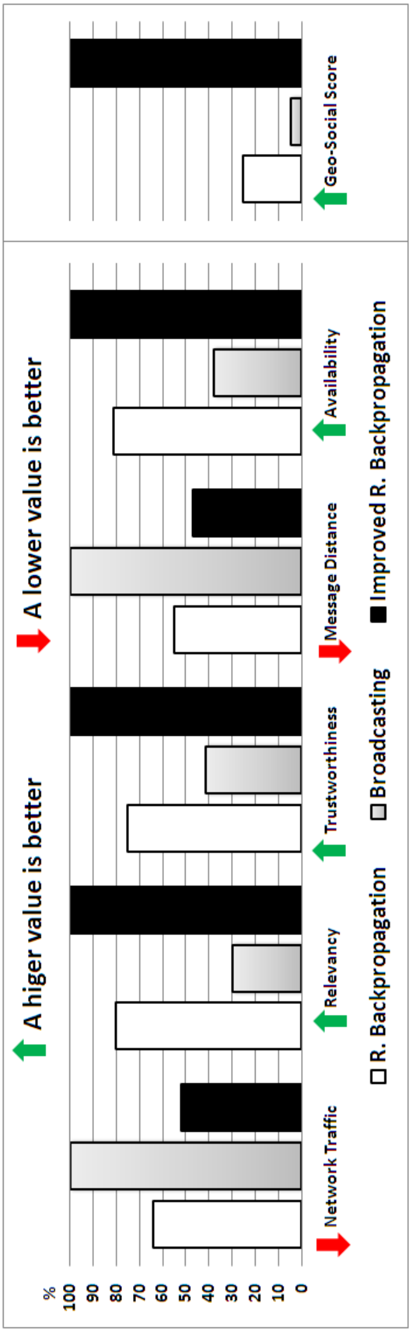


Figure 6.9. Improved relevance back propagation technique for routing messages in the network shows better results for each evaluated parameter.

Chapter 7

Conclusions and future work

In the beginning of this dissertation, we formulated four research challenges. Each challenge was accompanied by a research question. In this chapter we provide a summary of our findings and contributions with respect to each research question that the work presented in this thesis addressed. We then proceed with a discussion on possibilities for follow-up research and also share ideas for longer-term impact.

7.1. Summary and conclusions

In chapter 2 we gave an overview of the background in the area of pervasive location modelling and sensing. Keeping in mind the research questions formulated in chapter 1, the analysis and the discussion of the existing solutions and problems resulted in a set of requirements for a pervasive location model, addressed from the point of view of each category of the application's creators, who are its developers, designers and end users. In short, while the development and the systems aspects have been addressed by other researchers, our requirements were specified by looking at all of them together.

We now discuss how each chapter contributes to the corresponding research challenge and answers the accompanying research question.

7.1.1. RQ1: regarding uncertainty, easiness, openness

In short, the first research question, RQ1, can be summed up into three features that a location model we were looking for was expected to have: provide equally ***easy perception*** of the model's concepts by all parties, handle ***uncertainty***, and connect its own concepts to ***other context***. In chapter 3, we introduced a user-centric view on the spatial relations among resources in a pervasive environment and explained the view's underlying model. In relation to each feature, the model contributed as follows:

Easiness The model's straightforward and flexible representation of entities and spatial relationships among them as a semantically enriched graph, while already suiting developers (Glassey, 2009) as well as being helpful in designing spatially-aware user interfaces (Kortuem et al., 2005), provided the transparency of the model's concepts. The integration of the model's concepts into a framework for working with pervasive applications (Vanderhulst, 2010, chapter 5), coupled with using the natural language approach to representing the cardinal compass-like directions and corresponding relations therein, the compass's subsequent division into zones, and its visual representation in a dedicated spatial tool within that framework assured the representation's applicability for users. Any changes introduced into the model's representation, while initiated conceptually by designers and incorporated by developers, are immediately reflected in the framework's corresponding spatial arrangements' query plug-in, in which the same natural language notation within a predefined and explained format can be used.

Uncertainty The "node availability" and "node importance" extensions into the graph's nodes contributed to the location model's ability to coping with the uncertainty of a resource's behaviour during interaction. The combination of the interaction range and a fuzzy-based nearby extensions of the model's representation, while remaining visually transparent, allowed the framework to address and distinguish between subtle situations of the resources' spatial arrangements both visually and numerically. Again, any changes introduced into the model's representation are reflected in the framework's corresponding spatial arrangements' query plug-in.

Other context The implementation of the model and its concepts, both the basic ones and the extensions for handling uncertainty, as an ontology attached to the framework, together with the spatial model's established connecting links to the framework's core environment ontology, assured the location model's collaboration with other parts of a pervasive environment. The framework's ability to add and remove the required domain ontologies assured the spatial model's flexible communication with their context models through the framework's core functionality in an upon-request style.

7.1.2. RQ2: regarding uncertainty-aware ubiquitous localisation

Our second research challenge was to investigate the variability and uncertainty of the localisation process in a pervasive environment from the point of view of its ubiquity. The challenge's research question, RQ2, wondered, "How can we handle ***any available localisation***, reflect its ***uncertainty and limitations***, and do it in a way that would ***ease information understanding*** between the providers and the consumers of localisation data?" Furthermore, we have identified in chapter 2, section 2.4, a collection of requirements that such an approach would be expected to comply with: ***availability and scalability, flexibility***, and ***non-centralised processing***. In chapter 4 we proposed a unified view on uncertainty-aware ubiquitous localisation. The view's model addressed the properties and details of localisation systems and used this information to build an approach on addressing objects' localisation throughout a pervasive environment, so that the following was achieved:

Any localisation The proposed localisation ontology for modelling localisation systems' metadata and representing the details of each system's positioning process ensured that the data received from an arbitrary localisation system could be represented in one shared format, so that any location provider, i.e. available at a given moment, is interpreted as an equal player, as well as the absence of any provider is treated appropriately.

Uncertainty and limitations Using the description of the system's parameters provided during this system's inclusion phase, the uncertainty of this system's localisation process was reflected in the ontology's corresponding attributes, so that the behaviour details got appropriately addressed by the reasoning rules.

Availability and scalability The involved Global Sensor Networks platform, a middleware for diverse sensor data processing, helped to convert the initially diverse location data into the unified format of the localisation ontology. The use of W2P, an HTTP-based communication system, ensured easy and stable delivery and reception of location information.

Flexibility The use of GSN's both the "wrapper" methodology for data reception and the "virtual sensor" abstraction for data processing ensured that the variability of the (initially diverse) location data is handled correctly. In the specified notation, if a localisation system provides no information about a component (or, on the contrary, has an extra characteristic), the corresponding field is left empty (or, respectively, is created) during the data reception, and this situation is then treated appropriately during the processing step.

Non-centralised processing GSN's and W2P's (symmetric) peer-to-peer approach to information reception, exchange, and delivery made them both a suitable choice for our goal to decentralise access to location tracking provision, to consider both domestic localisation systems and global services, and to provide remote communication when required.

Easing information understanding The virtual sensor's XML notation, the GSN's ability to support the introduction of new location providers in a plug-and-play style at run-time, and the included W2P-based delivery channel – on the one hand, made the inclusion of a (generally unknown) arbitrary location provider less cumbersome and technology-specific from the point of view of location data producers. The interpretation of a localisation system's parameters within the localisation ontology and the inclusion of this ontology into the location context model with pre-available tools for exploring the ontology – on the other hand, made the access to the corresponding localisation system easier from the point of view of location data consumers.

7.1.3. RQ3: regarding user awareness of the variability of location context in a graphical user interface

The third research question that we posed, RQ3, was devoted to the incorporation of location context variability into a pervasive application's graphical user interface. In particular, we wondered which ***aspects*** of this variability are useful to know about, in which ***form*** they should be ***presented***, and in which ***situations***. In chapter 5 we described the test facilities, the technology, the conditions, the results, and the analysis of two user studies that we performed with respect to the above. Since the investigated concepts exist in close connection to each other, the answers outlined below are also presented as a chain of findings, and detail as follows:

Aspects, part 1 From the first user study, we were able to conclude that users' impressions about being aware of the changes in the active tracking are positive, and that users prefer to stay aware of both the uncertainty of their position estimate and the source of this uncertainty (i.e. the currently active location tracking system).

Form of presentation Based on Benford et al.'s (2006, p.122) classification of "states of being" of a mobile user, we introduced a set of states of awareness about the status of a mobile user's location that reflected the behaviour of an active localisation system and take into account uncertainty. Each state's visualisation involved a combination of independent elements – a dot, a circle, and a cross – that we referred to as visualisation patterns.

Aspects, part 2 Each visualisation pattern involved in presenting a user's location tracking conditions in the second user study – reflecting, respectively, the frequency of location updates, the error of a position estimate, and the absence of location updates – proved to be beneficial.

Situations A thorough evaluation of each visualisation pattern's performance revealed that their impacts depended on each user's personal profile (such as their eyesight level, navigation skills, ability to understand maps, etc.), distance between the users (knowing about finer details of someone else's state is less important if they are far), and the quality and reliability of the tracking. Based on these and other observations and conclusions, we were able to suggest a set of design guidelines and implications on visualising the corresponding aspects of the variability of location context in pervasive applications and their presentation to users, which included:

- The importance of the dot (given that it can be perceived if shown) increases in problematic areas where location tracking may become unreliable or absent, or in situations when participants are close to each other and consult the map more often;

- The circle's gradual extension while location tracking remained stable was less noticed. However, in the problematic areas, the extending circle that was gradually transforming and got replaced, after a time, with a cross gave the feeling of movement and dynamics and created a firm positive attitude;
- The cross for the case of missing tracking updates was important for the awareness and should always be activated.

7.1.4. RQ4: regarding location and social context in VANETs

As stated, the location context's existence in close connections with a variety of other aspects within an environment made us consider its use as a catalyst for the improvement of existing scenarios and situations. Therefore in our fourth research challenge's question, RQ4, we asked how ***location and social context*** can be ***combined*** in order to ***improve information filtering*** in a large-scale vehicular ad-hoc network (***VANET***). In chapter 6 we investigated a ubiquitous help-seeking/provision scenario, for which we introduced an approach that tied together information about vehicles' locomotion and drivers' social profiles. We evaluated its performance on a large-scale VANET's realistic data set that we fed into a real-time discrete event-based network simulator OMNET++. The following was produced and achieved:

Combining location and social context The introduced "geo-social relevance function" involved a number of each (vehicle-and-its-driver) pair's spatial and social characteristics. The function combined these characteristics in a balanced manner into a formula for calculating each driver's relevance score as a potential help provider to the help request being answered, with a higher score meaning this driver's better suitability to this particular request.

Improving information filtering in VANETs The comparison of the proposed approach to two previously known approaches to information dissemination in VANETs – namely, using a simple back propagation information filtering technique and the state-of-the-art case of plain broadcasting – revealed the approach's advantage in each of the analysed major network metrics, including network traffic, relevancy, trustworthiness, message distance, and availability. Besides, the average geo-social score of all selected help providers in all executed cases of help requests when using the proposed geo-social approach greatly outweighed the scores of the chosen help providers in both other approaches, for which the scores were calculated for the purpose of future reference and comparison but were not involved in information filtering, meaning the help providers' overall better suitability.

7.2. Future work

The niche of location context within the entire domain of pervasive computing is broad. This dissertation investigated only a small part of this niche, and the proposed answers and findings contributed to a great deal of efforts which have been, still are being, and are yet to be made in the area of location-aware computing. Below we outline some directions of applying and extending this dissertation's findings.

One aspect of further improvements from the technical point of view lies in extending the ways localisation systems can be integrated into the framework. Possible solutions include developing mapping components for the currently required Java-based wrapper or accomplishing mappings for the different localisation systems on the ontology level. Ontology mapping allows one to discover similarities and differences between systems automatically using an ontology reasoner such as Pellet, so exploring this opportunity seems interesting.

We addressed run-time location tracking and considered its omnidirectional and passive prediction. Recent studies have discussed that knowing about users' future locations is promising and have demonstrated the success of deliberate efforts on location prediction (e.g., Burbey, 2011; Scellato et al., 2011). Therefore an interesting extension lies in addressing visualisations for predicted short- and long-term locations. Besides, we displayed the location information of only a few users in our application. Prior research results have shown that users are able to pay attention to and track about four randomly moving objects (Pylyshyn, 1994). In this regard, additional visualisation patterns and strategies are likely required for multi-user applications.

Within the scope of this work, we advocated the ego-centric view on spatial relations and referred to other objects' locations using the referred object's frame of reference, such as being in front or on the right. It was a justified choice for applications requiring location-driven interaction with nearby or reachable entities. However, as the scale increases, the use of the ego-centric view becomes less straightforward. If, for example, a user is in Japan and needs to refer to Australia, it becomes reasonable to use an absolute coordinate system in this case and say that Australia is in the south, regardless of the user's orientation in space. Therefore, an investigation is necessary that will allow us to understand the range of the ego-centric view's applicability, possibly followed by a development of the spatial model for ultra-large-scale scenarios, where priorities of different categories of location information are likely to change.

The amount of location information users are interested in or willing to give is greatly affected by various details, often considered in combination with each other (Reilly et al., 2006). Therefore incorporating users' social profiles, their personal preferences or agendas into the adaptation model would help to single the relevant users out and to reduce the visualisations of the rest. In this regard, a closely related and very important aspect is to reflect location privacy, security and trust. Here, some results have been recently achieved by Tang et al. (2010, 2011), so that these works could possibly be used as starting references. But the topic's breadth, supported by studies on the diversity of mobile users' information needs and preferences therein (e.g., Dearman, Kellar, and Truong, 2008; Church et al., 2010), and its high impact on an application's overall success clearly indicate that additional investigations are welcome.

Bibliography

Abdelsalam, W., and Ebrahim, Y., 2004. A Bayesian-networks-based approach for managing uncertainty in location-tracking. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, volume 4, pages 2205–2208.

Aberer, K., Hauswirth, M., and Salehi, A., 2007. Infrastructure for data processing in large-scale interconnected sensor networks. In *Proceedings of the 8th International Conference on Mobile Data Management*, pages 198–205.

Anagnostopoulos, C.B., Ntarladimas, Y., and Hadjiefthymiades, S., 2007. Situational computing: an innovative architecture with imprecise reasoning. *Journal of System and Software*, 80(12), 1993–2014.

Aparicio, S., Perez, J., Bernardos, A.M., and Casar, J.R., 2008. A fusion method based on Bluetooth and WLAN technologies for indoor location. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems,* pages 487–491.

Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., and Samarati, P., 2007. Location privacy protection through obfuscation-based techniques. In *Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 47–60.

Arminen, I., 2006. Social functions of location in mobile telephony. *Personal and Ubiquitous Computing,* 10(5), 319–323.

Azizyan, M., Constandache, I., and Choudhury, R.R., 2009. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pages 261–272.

Backstrom, L., Sun, E., and Marlow, C., 2010. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, pages 61–70.

Bahl, P., and Padmanabhan, V.N., 2000. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual IEEE Conference on Computer Communications,* pages 775–784.

Banerjee, N., Agarwal, S., Bahl, P., Chandra, R., Wolman, A., and Corner, M., 2010. Virtual Compass: relative positioning to sense mobile social interactions. In *Proceedings of the 8th International Conference on Pervasive Computing,* pages 1–21.

Barkhuus, L., Brown, B., Bell, M., Sherwood, S., Hall, M., and Chalmers, M., 2008. From awareness to repartee: sharing location within social groups. In *Proceedings of the 26th Conference on Human Factors in Computing Systems*, pages 497–506.

Baudisch, P., and Rosenholtz, R., 2003. Halo: a technique for visualizing off-screen objects. In *Proceedings of the 21st Conference on Human Factors in Computing Systems,* pages 481–483.

Baus, J., and Kray, C., 2002. Frames of reference, positional information and navigational assistance. In *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*, pages 461–465.

Baus, J., Kray, C., and Krüger, A., 2001. Visualisation of route descriptions in a resource-adaptive navigation aid. *Cognitive Processing,* 2(2-3), 323–345.

Beigl, M., Zimmer, T., and Decker, C., 2002. A location model for communicating and processing of context. *Personal and Ubiquitous Computing,* 6(5-6), 341–357.

Bellavista, P., Corradi, A., and Giannelli, C., 2008. The PoSIM middleware for translucent and context-aware integrated management of heterogeneous positioning systems. *Computer Communications,* 31(6), 1078–1090.

Benford, S. et al., 2006. Can you see me now?. *ACM Transactions on Human-Computer Interaction,* 13(1), 100–133.

Bernheim Brush, A.J., Krumm, J., and Scott, J., 2010. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Proceedings of the 12th International Conference on Ubiquitous Computing*, pages 95–104.

Bettini, C., Brdzicka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D., 2010. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2), 161–180.

Bilandzic, M., Foth, M., and De Luca, A., 2008. CityFlocks: designing social navigation for urban mobile information systems. In *Proceedings of the 7th ACM Conference on Designing Interactive Systems*, pages 174–183.

Boesen, J., Rode, J.A., and Mancini, C., 2010. The domestic panopticon: location tracking in families. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, pages 65–74.

Bose, A., and Foh, Ch.-H., 2007. A practical path loss model for indoor WiFi positioning enhancement. In *Proceedings of the 6th International Conference on Information, Communications & Signal Processing,* pages 1–5.

Burbey, I., 2011. Predicting future locations and arrival times of individuals. *PhD thesis*. Virginia Polytechnic Institute and State University, USA.

Burgess, N., 2006. Spatial memory: how egocentric and allocentric combine. *Trends in Cognitive Science*, 10(12), 551–557.

Burigat, S. and Chittaro, L., 2011. Pedestrian navigation with degraded GPS signal: investigating the effects of visualizing position uncertainty. In *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 221–230.

Burigart S., Chittaro, L., and Gabrielli, S., 2006. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services,* pages 239–246.

Butz, A., Baus, J., Krüger, A., and Lohse, M., 2001. A hybrid indoor navigation system. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, pages 25-32.

Castro, P., and Muntz, R., 2000. Managing context data for smart spaces. *IEEE Personal Communications*, 7(5), 44–46.

Cearreta, I. and Garay-Vitoria, N., 2011. Toward adapting interactions by considering user emotions and capabilities. In *Proceedings of the 14th International Conference on Human-Computer Interaction,* pages 525–534.

Chen, H., Perich, F., Finin, T., Joshi, A., 2004. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 258–267.

Cheng, Y.-Ch., Chawathe, Y., LaMarca, A., and Krumm, J., 2005. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications and Services*, pages 233–245.

Church, K., Neumann, J., Cherubini, M. and Oliver, N., 2010. The "map trap"?: an evaluation of map versus text-based interfaces for location-based mobile search services. In *Proceedings of the 19th International Conference on World Wide Web*, pages 261–270.

Cohn, A.G., Bennett, B., Gooday, J., and Gotts, N.M., 1997. Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(3), 275–316.

Coronato, A., Esposito, M., and Pietro, G., 2009. A multimodal semantic location service for intelligent environments: an application for smart hospitals. *Personal and Ubiquitous Computing*, 13(7), 527–538.

Coyle, L., Ye, J., Loureiro, E., Knox, S., Dobson, S., and Nixon, P., 2007. A proposed approach to evaluate the accuracy of tag-based location systems. In *Proceedings of the Workshop on Ubiquitous Systems Evaluation,* pages 292–296.

Damián-Reyes, P., Favela, J., and Contreras-Castillo, J. 2011. Uncertainty management in context-aware applications: increasing usability and user trust. *Wireless Personal Communications,* 56(1), 37–53.

Dearman, D., Hawkey, K., and Inkpen, K., 2005. Rendezvousing with location-aware devices: enhancing social coordination. *Interacting with Computers,* 17(5), 542–566.

Dearman, D., Inkpen, K.M., and Truong, K.N., 2010. Mobile map interactions during a rendezvous: exploring the implications of automation. *Personal and Ubiquitous Computing*, 14(1), 1–13.

Dearman, D., Kellar, M., and Truong, K.N., 2008. An examination of daily information needs and sharing opportunities. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pages 679–688.

Dearman, D. and Truong K.N., 2010. Identifying the activities supported by locations with community-authored content. In *Proceedings of the 12th International Conference on Ubiquitous Computing,* pages 23–32.

Dearman, D., Varshavsky, A., de Lara, E., and Truong, K.N., 2007. An exploration of location error estimation. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, pages 181–198.

Dey, A. K., 2009. Context-aware computing. *In* J. Krumm, ed. *Ubiquitous computing fundamentals*. Boca Raton, FL: Chapman and Hall/CRC, 321–352.

Dey, A. K., and Abowd, G. D., 2000. Towards a better understanding of context and context-awareness. In *Workshop on the What, Who, Where, When and How of Context-Awareness*.

Dey, A.K., Hightower, J., de Lara, E., and Davies, N., 2010. Location-based services. *IEEE Pervasive Computing*, 9(1), 11–12.

Diestel, R., 2010. Graph Theory (4th ed.)*. Vol. 173 Graduate Texts in Mathematics.* Springer-Verlag.

Ding, Z., and Peng, Y., 2004. A probabilistic extension to ontology language OWL. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 1–10.

Dobson, S., 2005. Leveraging the subtleties of location. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence,* pages 189–193.

Dürr, F., and Rothermel, K., 2003. On a location model for fine-grained geocast. In *Proceedings of the 5th International Conference on Ubiquitous Computing,* pages 18–35.

Fischer, C., and Gellersen, H., 2010. Location and navigation support for emergency responders: a survey. *IEEE Pervasive Computing*, 9(1), 38–47.

Franklin, M., Jeffery, S., Krishnamurthy, S., Reiss, F., Rizvi, S., We, E., Cooper, O., Edakkunni, A., and Hong, W., 2005. Design considerations for high fan-in systems: the HiFi approach. In *CIDR 2005,* pages 290–304.

Franklin, N., and Tversky, B., 1990. Searching imagined environments. *Journal of Experimental Psychology: General,* 119(1), 63–76.

Fuchs, C., Aschenbruck, N., Martini, P., and Wieneke, M., 2011. Indoor tracking for mission critical scenarios: a survey. *Pervasive and Mobile Computing,* 7(1), 1–15.

Gellersen, H., Lukowicz, P., Beigl, M., and Riedel, T., 2010. Cooperative relative positioning. *IEEE Pervasive Computing,* 9(4), 78–89.

Geusgen, H., 2002. Reasoning about distance based on fuzzy sets. *Applied Intelligence,* 17(3), 265–270.

Gibbons, P.B., Karp, B., Ke, Y., Nath, S., and Seshan, S., 2003. IrisNet: an architecture for a world-wide sensor web. *IEEE Pervasive Computing, 2*(4), 22–33.

Girardin, F., 2007. Bridging the socio-technical gap in location-aware computing. In *Proceedings of the 25th Conference on Human Factors in Computing Systems Extended Abstracts,* pages 1653–1656.

Glassey, R., 2009. Location modelling and management in ubiquitous computing environments. *PhD thesis.* University of Strathclyde, UK.

Gray, P.D., and Salber, D., 2001. Modelling and using sensed context information in the design of interactive applications. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, pages 317–336.

Grolaux, D., Van Roy, P., and Vanderdonckt, J., 2004. Migratable user interfaces: beyond migratory interfaces. In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services,* pages 422–430.

Gruber, T., 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies,* 43(5-6), 907–928.

Gu, T., Pung, H.-K., and Zhang, D.-Q., 2004. Toward an OSGi-based infrastructure for context-aware applications. *IEEE Pervasive Computing*, 3(4), 66–74.

Gu, T., Pung, H.-K., and Zhang, D.-Q., 2004b. A Bayesian approach for dealing with uncertain contexts. In *Proceedings of Advances of the 2nd International Conference on Pervasive Computing*, pages 205–210.

Gu, H.-M., Wang, X., Ling, Y., and Shi, J.-Q., 2007. Building a fuzzy ontology of edutainment using OWL. In *Proceedings of the 7th International Conference on Computational Science,* pages 591–594.

Hansen, R., Wind, R., Jensen, C.S., and Thomsen, B., 2009. Seamless indoor/outdoor positioning handover for location-based services in Streamspin. In *Proceedings of the 10th International Conference on Mobile Data Management*, pages 267–272.

Harter, A., Hopper, A., Steggles, P., Ward, A., and Webster, P., 2002. The anatomy of a context-aware application, *Wireless Networks* 8(2–3), 187–197.

Heckerman, D., 1996. A tutorial on learning with Bayesian networks. *Technical Report MSR-TR-95-06*. Microsoft Research, US.

Hervás, R. and Bravo, J., 2011. Towards the ubiquitous visualization: Adaptive user-interfaces based on the Semantic Web. *Interacting with Computers* 23(1), 40–56.

Hightower, J., and Boriello, G., 2001. Location systems for ubiquitous computing. *Computer,* 34(8), 57–66.

Hightower, J., Brumitt, B., and Boriello, G., 2002. The Location Stack: a layered model for location in ubiquitous computing. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications,* pages 122–128.

Hightower, J., LaMarca, A., and Smith, I., 2006. Practical lessons from Place Lab. *IEEE Pervasive Computing,* 5(3), 32–39.

Hii, P., and Zaslavsky, A., 2005. Improving location accuracy by combining WLAN positioning and sensor technology. In *Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks*.

Hoffmann-Wellenhof, B., Lichtenegger, H., and J. Collins., 1992. Global Positioning System: Theory and Practice (3rd ed.). Springer-Verlag.

Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., and Schwehm, M., 1999. Next century challenges: Nexus – an open global infrastructure for spatial-aware applications. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 249–255.

Hu, H., and Lee, D.L., 2004. Semantic location modeling for location navigation in mobile environment. In *Proceedings of the 5th IEEE International Conference on Mobile Data Management*, pages 52–61.

Iachini, T., and Ruggiero, G., 2006. Egocentric and allocentric spatial frames of reference: a direct measure. *Cognitive Processing,* 7(1), 126–127.

Jiang, C., and Steenkiste, P., 2002. A hybrid location model with a computable location identifier for ubiquitous computing. In *Proceedings of the 4th International Conference on Ubiquitous Computing,* pages 246–263.

Judd, G., and Steenkiste, P., 2003. Providing contextual information to pervasive computing applications. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 133–142.

Kay, J., Niu, W., and Carmichael, D.J., 2007. ONCOR: ontology- and evidence-based context reasoner. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 290-293.

Kenteris, M., Gavalas, D., and Economou, D., 2011. Electronic mobile guides: a survey. *Personal and Ubiquitous Computing,* 15(1), 97–111.

Kjærgaard, M.B., Blunck, H., Godsk, T., Toftkjær, T., Christensen, D.L., and Grønbæk, K., 2010. Indoor positioning using GPS revisited. In *Proceedings of the 8th International Conference on Pervasive Computing,* pages 38–56.

Knauth, S., Kistler, R., and Klapproth, A., 2009. Towards a spatial position management system for interactive building environments. In *Proceedings of the International Conference on Intelligent Interactive Assistance and Mobile Multimedia Computing,* pages 201–211.

Koile, K., Tollmar, K., Demirdjian, D., Shrobe, H., and Darrell, T., 2003. Activity zones for context-aware computing. In *Proceedings of the 5th International Conference on Ubiquitous Computing,* pages 90–106.

Kortuem, G., Kray, C., and Gellersen, H., 2005. Sensing and visualizing spatial relations of mobile devices. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology,* pages 93–102.

Kurschl, W., Gottesheim, W., Mitsch, S., Prokop, R., Schonbock, J., and Beer, W., 2008. Towards a unified location tracking system for heterogeneous industrial environments. In *Proceedings of the 6th IEEE International Conference on Industrial Informatics,* pages 1267–1272.

Labrou, Y., and Finin, T., 1999. Yahoo! as an ontology: using Yahoo! categories to describe documents. In *Proceedings of the 8th International Conference on Information and Knowledge Management,* pages 180–187.

LaMarca, A. et al., 2005. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of the 3rd International Conference on Pervasive Computing*, pages 116–133.

Langdal, J., Kjærgaard, M.B., Toftkjær, T., and Schougaard, K., 2010a. Exposing position uncertainty in middleware. In *Proceedings of the 2nd International Workshop on Middleware for Pervasive Mobile and Embedded Computing,* pages 1–8.

Langdal, J., Schougaard, K.R., Kjærgaard, M.B., and Toftkjær, T., 2010b. PerPos: a translucent positioning middleware supporting adaptation of internal positioning processes. In *Proceedings of the 11th International Conference on Middleware,* pages 232–251.

Lei, H., Sow, D.M., Davis, J.S.II, Banavar, G., and Ebling, M.R., 2002. The design and applications of a context service. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(4), 45–55.

Lemelson, H., King, T., and Effelsberg, W., 2008. A study on user acceptance of error visualization techniques. In *Proceedings of the International Workshop on Human Control of Ubiquitous Systems.*

Lemelson, H., Kjærgaard, M.B., Hansen, R., and King, T., 2009. Error estimation for indoor 802.11 location fingerprinting. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, pages 138–155.

Leonhardt, U., 1998. Supporting location-awareness in open distributed systems. *PhD thesis.* University of London, UK.

Li, N., and Chen, G., 2009. Multi-layered friendship modeling for location-based Mobile Social Networks. In *Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 1–10.

Lim, B., and Dey, A., 2011. Investigating intelligibility for uncertain context-aware applications. In *Proceedings of the 13th International Conference on Ubiquitous Computing,* pages 415–424.

Lorenz, B., Ohlbach, H.J., and Stoffel, E.-P., 2006. A hybrid spatial model for representing indoor environments. In *Proceedings of the 6th International Conference on Web and Wireless Geographical Information Systems*, pages 102–112.

Lovett, T., O'Neill, E., Irwin, J., and Pollington, D., 2010. The calendar as a sensor: analysis and improvement using data fusion with social networks and location. In *Proceedings of the 12th International Conference on Ubiquitous Computing,* pages 3–12.

Lukowicz, P., Pentland, A., and Ferscha, A., 2012. From context awareness to socially aware computing. *IEEE Pervasive Computing,* 11(1), 32–41.

Manasseh, C., Ahern, K., and Sengupta, R., 2009. The Connected Traveler: using location and personalization on mobile devices to improve transportation. In *Proceedings of the 2nd International Workshop on Location and the Web*.

Massink, M., and Faconti, G., 2002. A reference framework for continuous interaction. *Universal Access in the Information Society,* 1(4), 237–251.

Matic, A., Papliatseyeu, A., Osmani, V., and Mayora-Ibarra, O., 2010. Tuning to your position: FM-radio based indoor localization with spontaneous recalibration. In *Proceedings of the 8th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 153–161.

McNamara, T., Rump, B., and Werner, S., 2003. Egocentric and geocentric frames of reference in memory of large-scale space. *Psychonomic Bulletin & Review,* 10(3), 589–595.

Misund, G., Holone, H., Tolsby, H., Karlsen, J., and Toppe, A., 2009. Effects of co-player visualization in a location-based chase-and-catch game. In *Proceedings of the 11th International Conference on Human Computer Interaction with Mobile Devices and Services*.

Nakamura, K., and Shinoda, H., 2010. High accuracy position and orientation detection in two-dimensional communication network. In *Proceedings of the 28th Conference on Human Factors in Computing Systems*, pages 2297–2306.

Niu, W., and Kay, J., 2008a. Location conflict resolution with an ontology. In *Proceedings of the 6th International Conference on Pervasive Computing*, pages 162–179.

Niu, W., and Kay, J., 2008b. Pervasive personalisation of location information: personalised context ontology. In *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 143–152.

Nova, N., Girardin, F., and Dillenbourg, P., 2010. The effects of mutual location-awareness on group coordination. *International Journal of Human-Computer Studies,* 68(7), 451–467.

Nurmi, P., and Bhattacharya, S., 2008. Identifying meaningful places: the non-parametric way. In *Proceedings of the 6th International Conference on Pervasive Computing*, pages 111–127.

Olmsted-Hawala, E.L., Murphy, E., Hawala, S., and Ashenfelter, K.T., 2010. Think-aloud protocols: a comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. In *Proceedings of the 28th Conference on Human Factors in Computing Systems*, pages 2381–2390.

Ongenae, M., et al., 2008. Design of a semantic person-oriented nurse call management system. *International Journal of Web and Grid Services*, 4(3), 267–283.

Oppermann, L., 2009. An abstract location model for mobile games. In *GI Jahrestagung*, pages 1863–1872.

Opperman, L., Broll, G., Capra, M., and Benford, S., 2006. Extending authorizing tools for location-aware applications with an infrastructure visualization layer. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, pages 52–68.

Patel, S.N., Kientz, J.A., and Gupta, S., 2010. Studying the use and utility of an indoor location tracking system for non-experts. In *Proceedings of the 8th International Conference on Pervasive Computing*, pages 228–245.

Patel, S., Truong, K.N., and Abowd, G., 2006. PowerLine positioning: a practical Sub-room-level indoor location system for domestic use. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, pages 441–458.

Patricia, 2010. Average human walking speed: walking for racing | race walking | speed of walking [online]. Yoga Health Benefits. Available from: http://www.yogawiz.com/blog/walking/average-human-walking-speed.html [accessed 5 December 2011].

Peddinti, S.T., Dsouza, A., and Saxena, N., 2011. Cover locations: availing location-based services without revealing the location. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, pages 143–152.

Popleteev, A., 2011. Indoor positioning using FM radio signals. *PhD thesis.* University of Trento, Italy.

Preuveneers, D. et al., 2004. Towards an extensible context ontology for ambient intelligence. In *Proceedings of the 2nd European Symposium on Ambient Intelligence,* pages 148–159.

Pulkkinen, T., Bhattacharya, S., and Nurmi, P., 2011. Abstracting positions for indoor location based services. In *Proceedings of the IUI Workshop on Location Awareness for Mixed and Dual Reality.*

Pylyshyn, Z., 1994. Some primitive mechanisms of spatial attention. *Cognition*, 50, 363–384.

Ranganathan, A., Al-Muhtadi, J., and Campbell, R., 2004c. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2), 62–70.

Ranganathan, A., Al-Muhtadi, J., Chetan, S., Campbell, R., and Mickunas, D., 2004a. MiddleWhere: a middleware for location-awareness in ubiquitous computing applications. In *Proceedings of the 5th International Conference on Middleware,* pages 397–416.

Ranganathan, A., McGrath, R. E., Campbell, R. H. and Mickunas, M. D., 2004b. Use of ontologies in a pervasive computing environment. *Knowledge Engineering Review*, 18(3), 209–220.

Raney, B., Voellmy, A., Cetin, N., Vrtic, M., and Nagel, K., 2002. Towards a microscopic traffic simulation of all of Switzerland. In *Proceedings of the International Conference on Computational Science*, pages 371–380.

Ravi, N., Shankar, P., Frankel, A., Elgammal, A., and Iftode, L., 2006. Indoor localization using camera phones. In *Proceedings of the 7th IEEE Workshop on Mobile Computing Systems and Applications*, pages 1–7.

Reichenbach, F., and Timmermann, D., 2006. Indoor localization with low complexity in wireless sensor networks. In *Proceedings of the IEEE International Conference on Industrial Informatics,* pages 1018–1023.

Reilly, D., Dearman, D., Ha, V., Smith, I., and Inkpen, K., 2006. "Need to know": examining information need in location discourse. In *Proceedings of the 4th International Conference on Pervasive Computing,* pages 33–49.

Ruggiero, G., and Iachini, T., 2006. The effect of familiarity on egocentred and allocentred spatial representations of the environment. *Cognitive Processing,* 7(1), 88–89.

Satoh, I., 2007. A location model for smart environments. *Pervasive and Mobile Computing,* 3(2), 158–179.

Scellato, S., Musolesi, M., Mascolo, C., Latora, V., and Campbell, A., 2011. NextPlace: a spatio-temporal prediction framework for pervasive systems. In *Proceedings of the 9th International Conference on Pervasive Computing,* pages 152–169.

Schilit, B., and Theimer, M., 1994. Disseminating active map information to mobile hosts. *IEEE Network,* 8(5), 22–32.

Scipioni, M.P., and Langheinrich, M., 2010. I'm here! Privacy challenges in mobile location sharing. In *Proceedings of the 2nd International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use*.

Sentz, K., and Ferson, S., 2002. Combination of evidence in Dempster-Shafer theory. *Technical Report SAND 2002-0835*. Sandia National Laboratories, US.

Shen, X., Wang, Z., Jiang, P., Lin, R., and Sun, Y., 2005. Connectivity and RSSI based localization scheme for wireless sensor networks. In *Proceedings of the 2005 International Conference on Intelligent Computing,* pages 578–587.

Shneidman, J., Pietzuch, P., Ledlie, J., Roussopoulos, M., Seltzer, M., and Welsh, M., 2004. Hourglass: an infrastructure for connecting sensor networks and applications. *Technical report TR-21-04.* Harvard University, US. Available online at ftp://ftp.deas.harvard.edu/techreports/tr-21-04.pdf [accessed 21 Feb 2012].

Stevenson, G., Knox, S., Dobson, S., and Nixon, P., 2009. Ontonym: a collection of upper ontologies for developing pervasive systems. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, article no. 9.

Stevenson, G., Ye, J., Dobson, S., and Nixon, P., 2010. LOC8: A location model and extensible framework for programming with location. *IEEE Pervasive Computing*, *9*(1), 28–37.

Strang, T., and Linnhoff-Popien, C., 2004. A context modeling survey. In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management.

Streefkerk, J.W., van Esch-Bussenmakers, M.P., and Neerincx, M.A., 2008. Field evaluation of a mobile location-based notification system for police officers. In *Proceedings of the 10th international Conference on Human Computer Interaction with Mobile Devices and Services*, pages 101–108.

Strobbe, M. et al., 2007. Design of CASP: an open enabling platform for context aware office and city services. In *Proceedings of the 4th International Workshop on Managing Ubiquitous Communications and Services,* pages 123–142.

Strobbe, M. et al., 2012. Novel applications integrate location and context information. *IEEE Pervasive Computing*, *11*(2), 64–73.

Stuntebeck, E.P., Patel, S., Robertson, T., Reynolds, M.S., and Abowd, G.D., 2008. Wideband powerline positioning for indoor localization. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pages 94–103.

Tang, K., 2010. "Sometimes less is more": multi-perspective exploration of disclosure abstractions in location-aware social mobile applications. *PhD thesis.* Carnegie Mellon University, USA.

Tang, K., Hong, J., and Siewiorek, D.P., 2011. Understanding how visual representations of location feeds affect end-user privacy concerns. In *Proceedings of the 13th International Conference on Ubiquitous Computing,* pages 207–216.

Truong, B.A., Lee, Y.-K., and Lee, S.-Y., 2005. Modeling uncertainty in context-aware computing. In *Proceedings of the 4th Annual ACIS International Conference on Computer and Information Science*, pages 676–681.

Vanderhulst, G., 2010. Development and deployment of interactive pervasive applications for ambient intelligent environments. *PhD Thesis.* Hasselt University, Belgium.

Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., and Otsason, V., 2007. GSM indoor localization. *Pervasive and Mobile Computing*, 3(6), 698–720.

Varshavsky, A. and Patel, S., 2009. Location in ubiquitous computing. *In* J. Krumm, ed. *Ubiquitous computing fundamentals*. Boca Raton, FL: Chapman and Hall/CRC, 285–320.

Wagner, J. and Kray, C., 2010. Opportunistic peer-to-peer pedestrian positioning system. In *Poster at the 8th International Conference on Pervasive Computing.*

Want, R., Hopper, A., Falcao, V., and Gibbons, J., 1992. The active badge location system. *ACM Transactions on Information Systems*, 10(1), 91–102.

Weiser, M., 1991. The computer for the 21st century. *Scientific American*, 265(3), 66–75.

Widyawan et al., 2011. Virtual lifeline: multimodal sensor data fusion for robust navigation in unknown environments. *Pervasive and Mobile Computing*, online version, DOI: 10.1016/j.pmcj.2011.04.005 [accessed 23 Feb 2012].

Ye, J., 2009. Exploiting semantics with situation lattices in pervasive computing. *PhD thesis.* University College Dublin, Ireland.

Ye, J., Coyle, L., Dobson, S., and Nixon, P., 2007a. A unified semantics space model. In *Proceedings of the 3rd International Symposium on Location and Context Awareness*, pages 103–120.

Ye, J., Coyle, L., Dobson, S., and Nixon, P., 2007b. Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review,* 22(4), 315–347.

Ye, J., Coyle, L., Dobson, S., and Nixon, P., 2008b. Representing and manipulating situation hierarchies using situation lattices. *Revue d'Intelligence Artificielle*, 22(5), 647–667.

Ye, J., Dobson, S., and Nixon, P., 2008. An overview of pervasive computing systems. *In* K. Delaney, ed. *Ambient intelligence with microsystems: augmented materials and smart objects*. Springer, 3–17.

Ye, J., Stevenson, G., and Dobson, S., 2011. A top-level ontology for smart environments. *Pervasive and Mobile Computing,* 7(3), 359–378.

Zadeh, L., 1965. Fuzzy sets. *Information and Control,* 8(3), 338–353.

De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten uit twee landen: de Universiteit Hasselt en Maastricht University.
De opleidingen informatica/kennistechnologie, statistiek en biomedische wetenschappen/moleculaire levenswetenschappen zijn reeds ondergebracht in dit samenwerkingsverband. De bacheloropleiding in de rechten is een gezamenlijk initiatief van de Universiteit Hasselt, Maastricht University en de K.U.Leuven. Ook in andere wetenschapsdomeinen wordt gezocht naar samenwerking met andere universiteiten.

www.unimaas.nl

Universiteit Maastricht
Postbus 610
NL-6200 MD Maastricht
Tel.: 0031(0)43 388 222

www.uhasselt.be

Universiteit Hasselt | Campus Diepenbeek
Agoralaan | Gebouw D
BE-3590 Diepenbeek | België
Tel.: +32(0)11 26 81 11

The Variability of Location Context in Pervasive Environments: Modelling, Representation and Visualisation

Petr AKSENOV

DOCTORAATSPROEFSCHRIFT

2012 | School voor Informatietechnologie
Kennistechnologie, Informatica, Wiskunde, ICT

The Variability of Location Context in Pervasive Environments: Modelling, Representation and Visualisation

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, Informatica, te verdedigen door:

Petr AKSENOV

Promotor: prof. dr. Karin Coninx
Copromotor: prof. dr. Kris Luyten

Maastricht University

universiteit hasselt