A Survey and Taxonomy of Pervasive Computing Research

Chuong Cong Vo

Abstract

This work presents a survey and taxonomy of pervasive computing research. This work is aim to serve as a guideline for those who are new to pervasive computing and want to contribute to this research area.

1 Introduction

This work presents a classification of research areas on the pervasive computing paradigm. This work is aim to serve as a guideline for those who are new to pervasive computing and want to contribute to this research area.

2 Overview

2.1 What is pervasive computing?

Pervasive computing environments (or ubiquitous computing environments) are the environments into which computation is embedded. Computers and their applications are seamlessly integrated into our daily lives. Everyday devices can communicate with the human and with each others. They support nomadic users, mobile data access. They are "smart" and "active" spaces. They include context-aware applications. They provide anytime and anywhere access to information services while making the presence of the system "invisible" to the user.

Pervasive computing (or ubiquitous computing) was probably first used by Mark Weiser Weiser (1991). It is a paradigm of seamlessly integration of computational capabilities and information which are embedded in pervasive computing environments. Pervasive computing focuses on finding solutions to use pervasive environments more effectively and productively while make the availability of computers non-intrusive and virtually invisible to the user.

2.2 Objectives of pervasive computing

Weiser (1991, 1994) introduce the pioneer visions on ubiquitous computing. He asserted that the objective of ubiquitous computing is ubiquity (pervasive) and invisibility (autonomously).

Abowd and Mynatt (2000) argue that the implicit goal of this paradigm is "assisting everyday life and not overwhelming it".

Kang and Pisan (2006) show that one of the principal goals of pervasive computing is to be user-centric, providing computational support to help and mediate user activities in a natural, non-intrusive way.

By me
The general objective is to create a computing environment which meets, enhances, and supports human's daily activities (in general, human's life). The features of this environment evolve:

1. Adjust the environment to meet the user's preferences such as making a room suitable for the user's relaxing time.
2. Automatically (or semi-automatically) perform user's desired tasks which meets the user's intentions and in a non-disruption manners.
3. Offer tasks the user may want/intend to do and helping the user to accomplish the selected tasks by using services, devices, and resources which the user may not be aware of.

2.3 Characteristics of pervasive computing environments

1. Heterogeneity of devices such as devices' capabilities, platforms, and network protocols;
2. Unanticipated (unpredictable) availability of devices, network connections, and services;
3. Rapidly aging, spontaneously increase, ambiguities, incompleteness and uncertainty of context information;
4. Spontaneous mobility of devices and users which can lead to unavailability of devices/services;
5. Diverse interactions: (1) between users and the environment; (2) among multiple environmental resources; (3) among multiple users; (4) among small social groups(e.g., family, friend); (5) different combination of those types;
6. User preferences are invisible and seamlessly shift over time;
7. Preference conflictions
8. computational resources, battery power, memory, bandwidth limited and various.
9. immense amount of distributed system elements,Estrin et al. (2002)
10. Users in pervasive computing environments can be mobile and have computing sessions distributed over a range of devicesHenricksen et al. (2001)
11. Human attention is an especially scarce resource in such environments, because the user is often preoccupied with walking, driving, or other real-world interactions.Garlan et al. (2002)

2.4 Requirements for pervasive computing systems

2.4.1 by Huang et al. (1999)

- pervasive computing relies on infrastructure support: dependence is for communication and infrastructure services

2.5 byHenricksen et al. (2001)

- Mobility and distribution of software - Context awareness - Adaptation - Interoperability (devices, software, network) - Component discovery - tools supporting for rapid development and deployment applications - Scalability (due to increasing ubiquity of devices and software) - fault-tolerant - The infrastructures role with respect to users should be to maintain knowledge of their context (bao gom ca sessions) and to manage tasks related to their mobility - universally available user interfaces –¿ Adaptation to devices

in short - the ability to dynamically discover and compose software components in frequently changing environments - the ability to support increasingly autonomous and invisible applications through the provision of rich context information that is gathered from a wide range of sources, interpreted, and disseminated in a scalable fashion to interested parties - the ability to rapidly develop and deploy flexible software components that are adaptive and context-aware and, additionally, satisfy special requirements such as scalability and fault-tolerance - the ability to integrate heterogeneous computing environments, which have differing communication protocols and services (such as discovery mechanisms), into coherent pervasive computing systems that enable the formation of dynamic interactions between components - the ability to construct novel types of user interfaces that are universally available, regardless of the input and output capabilities of the available devices, that are sensitive to situation, and that are non-distracting

hay noi gon hon; - A scalable supporting infrastructure enable the dynamic discovery of software components and information; - the dynamic interconnection of components - the sensing, interpretation and dissemination of context; - the mobility and adaptation of components - the rapid development and deployment of large numbers of software components and user interfaces

2.5.1 by Saha and Mukherjee (2003)

- Scalability - cope with Heterogeneity - Integration: Though pervasive computing components are already deployed in many environments, integrating them into a single platform is still a research problem. - Invisibility: A system that requires minimal human intervention offers a reasonable approximation of invisibility. - Perception: Context awareness - Smartness: Context management (the effective use of context)

1. Security and privacy issues

2. Build Adaptable systems with minimum user interference

3. Provide personalised services to users

4. adaptation. Co 3 chien luoc cua adaptation: laissez-faire approach and application-transparent adaptation va ket hop ca hai.

2.6 Challenges of pervasive computing

2.6.1 Unsuitable technologies for pervasive computing Grimm et al. (2004)

such as Jini and Java RMI:

- A statically configured infrastructure: Name server, discovery server;
- A well-behaved computing environment: Transparent and synchronous invocations, no isolation between objects, no independence between devices;

2.7 Research areas of pervasive computing

2.7.1 Interconnectivity and interoperability

De xuat cac mo hinh giao tiep sao cho cac thiet bi hay dich vu de chung co the giao tiep voi nhau voi dieu kien la chung khong can phai biet truoc lan nhau khi chung duoc tao ra. Dieu nay la de dam bao cac thiet bi hay dich vu tuong tac, hop tac voi nhau trong moi truong pervasive. Mot moi truong ma thiet bi va dich vu lien tuc ra doi va doi hoi cac thiet bi hay dich vu truoc do phai giao tiep duoc voi chung. Telephone is an example of interoperability. It can communicate with cellular phones that did not exist at the time the rotary phone was built Edwards and Grinter (2001).

Cac nghien cuu ve van de nay:

- A common tuple space protocol Fox et al. (2000);
- Using HTTP protocols and content encodings (HTML) for interaction **?**;
- Using a set of interfaces that leverage mobile code to extend the behaviour of entities in the environment Edwards et al. (2001).
- uniform communication languages and standardised ontologies: (introduced by Kang and Pisan (2006))

2.7.2 Administration-free systems

- The "intelligence" of systems resides in the network itself such as telephone and cable television networks Edwards and Grinter (2001).
- "Outsourced" home administration, design technical solutions for remote diagnosis, administration, and software upgrades while with the security to prevent the kid next door from performing his own, unwarranted, remote diagnosis, administration, and upgrades Edwards and Grinter (2001).

- The bulk of functionality is placed in the network, not in the device itself Edwards and Grinter (2001); In the telephone system, for example, the telephone itself is the least complicated part of the system. And yet it provides access to new functionality available through the network without an upgrade or patch. This is a "utility" approach, in which the client technologies are shielded from upgrades and enhancements in the network, and yet can take advantage of new functionality when available.

2.8 Reliability

- Redundancy–data and services are replicated and available on multiple machines. Such an approach may, however, trade off against the goals of simplicity, intelligibility, and ease of administration Edwards and Grinter (2001).

2.9 Reasons of failure of pervasive computing

Banavar et al. (2000): not by devices, networking technologies, programming standards and APIs, but the notions people have of computing devices, applications and environment.

Warren (2004): - not enough time; - technology has not yet quite developed far enough; - price is high.

Me: the reason for this is the lack of an appropriately fundamental model of pervasive computing environments.

2.10 Visions of pervasive computing

By others:

- mobile computing devices as mini-desktops
- applications run on devices and exploit devices' functionalities.
- environment is a virtual space.

By Banavar et al. (2000):

- mobile computing devices as portals into applications/data spaces, not a repository of custom software managed by users.
- applications are means by which user performs a task.
- environment is user's information.

By Henricksen et al. (2001): - device
- users
- software component
- user interface

2.11 Recent development of technologies and society which impacts on pervasive computing

By Davies and Gellersen (2002):

- Processing power
- storage
- network
- GPS
- smart card
- RFID
- increasingly widespread acceptance of video surveillance in public places
- The World Wide Web
- digital mobile communications, digital mobile telephony
- Modern mobile phones with many capabilities and functions, low prices
- wireless networking, processing capability, storage capacity, and high-quality displays.

3 Classification

3.1 Overview and Visions

Weiser (1991, 1994) introduce the pioneer visions on ubiquitous computing. He asserted that the objective of ubiquitous computing is ubiquity (pervasive) and invisibility (autonomously).

Pascoe et al. (1999) present issues in developing context-aware computing.

Abowd and Mynatt (2000) argue that the implicit goal of this paradigm is "assisting everyday life and not overwhelming it".

Banavar et al. (2000); Davies and Gellersen (2002); Edwards and Grinter (2001); Kang and Pisan (2006); **?**); **?**); **?** identify visions and challenges.

Estrin et al. (2002); Garlan et al. (2002); Goslar and Schill (2004); **?** present characteristics of pervasive computing environments including *heterogeneity*, *individuality*, *offline use*, *indirect addressing*, *many applications*, and *community*.

Grimm et al. (2004) outline requirements for applications in pervasive computing environments.

Kang and Pisan (2006) show that one of the principal goals of pervasive computing is to be user-centric, providing computational support to help and mediate user activities in a natural, non-intrusive way.

3.2 Pervasive Scenarios

Home automation scenarios (Jazayeri, 2002): Mark arrives at home, tired after a day's work. The house identifies him and 'Unlocks the door'. Recognising the fatigue on his face, the home entertainment system starts 'Playing the favourite relaxing music', and 'Replaces the artwork on the electronic wallboards with soothing images of nature'. As he enters the bathroom, he is asked 'Warm the bath'. As he enters the kitchen, the display on the refrigerator door suggests 'Have a light meal'. As he accepts the suggestion, the system offers a task 'Cook a specific recipe' which is on the basis of the available ingredients in the refrigerator and fitting his mood. If he accepts the suggestion, step-by-step instructions are displayed, helping him through the steps of the recipe. Once he places the bowl on the stove, the system adjusts the temperature automatically to control the cooking time on the basis of his decision whether to take a bath.

Automotive scenario (Jazayeri, 2002): Bob is driving to a distant area for a two-week vacation. Bob plans to explore the area by car during his stay. On the way, an intelligent system detects an irregularity in the operation of the motor. It 'Flashes a warning sign' to him cautioning him to drive more carefully and 'Starts a diagnostic procedure'. It detects the offending part and 'Sends a report to his car manufacturer'. The manufacturer locates the garage nearest to his location and dispatches the part to the garage from a nearby depot. The garage sends a replacement car to meet him on the highway. The system 'Directs him to a specific parking area' off the highway to meet the driver that delivers his replacement car. He continues on his vacation and the driver takes his car to the garage. Once the car is repaired, it will be delivered to his hotel, using the information from his itinerary posted by his travel agent, aided by the positioning system in his car.

Health scenarios (Jackson, 2006): A health monitor, possibly installed in a watch, notices that Bob's blood sugar is suddenly elevated. It starts to 'Monitor other conditions' more closely to uncover possible causes and potential problems. It 'Contacts his medicine cabinet' to ensure that he has adequate medicine in his home and 'Contacts the pharmacy for additional medicine' if necessary. It 'Reports the change to his electronic medical record' and 'Sends a short message to his doctor', informing the doctor of your current location. Back at home, it 'Notifies the refrigerator to mark all sweets' in his home off-limit for him and suggests 'Cook a healthy meal' for the evening!

Health scenarios (Jackson, 2006): David notices a message on the heads-up display attached to his glasses, 'Lower room temperature'. He agrees that his kitchen feels a little warm, so he accepts. He then drives his wheelchair onto the back porch to watch his dog romp in the yard, but before he gets there, a soft alarm goes off and a message pops up on his display,'Watch The Braves game in five minute'. He glads to accept. His wheelchair automatically 'Drives into the living room', 'Turns on the big-screen TV' to the right channel, and 'Closes the drapes' for better viewing. The wheelchair system suggests, 'Invite Sam'. David accepts, and the wheelchair quickly 'Sends Sam a text message' that the game is starting and invites him to join David.

Official and Industrial workplace scenarios (Garlan et al., 2002): Fred is in his office, frantically preparing for a meeting at which he will give a presentation and a software demonstration. The meeting room is a 10-minute walk across campus. It is time to leave, but Fred is not quite ready. He grabs his PalmXXII wireless handheld computer and walks out of the door. Aura (an intelligent system) 'Transfers his work from his desktop to his handheld', and lets him make his final edits using voice commands during his walk. Aura infers where Fred is going from his calendar and the campus location tracking service. It 'Downloads the presentation and the demonstration software to the projection computer' and 'Warms up the projector'. Fred finishes his edits just before he enters the meeting room. As he walks in, Aura 'Transfers his final changes to the projection computer'. As the presentation proceeds, Fred is about to display a slide with highly sensitive budget information. Aura senses that this might be a mistake: the room's face detection and recognition capability indicates that there are some unfamiliar faces present. It therefore 'Warns Fred to skip this slide'. Realising that Aura is right, Fred skips that particular slide. He moves on to other topics and ends on a high note, leaving the audience impressed by his polished presentation.

Official and Industrial workplace scenarios (Luther et al., 2008): Two travellers, Dawson and his boss Fiona, arrive on a Friday morning at the Tokyo main station. Gordon, a project partner, is already waiting for them at the platform. The group is looking for a quick transfer to the airport. After having passed the gate at Tokyo station, Dawson's phone displays a basic list of recommended tasks such as 'Take the next train', 'Buy souvenirs', and 'Meet someone' because it has recognised that he has just arrived at the Tokyo station. He selects 'Meet someone'... The system also infers that Dawson and Fiona are colleagues travelling together. In addition, it reveals that the scene takes place at a weekday's morning at a public place during office hours. Fiona's phone shows a corresponding list of filtered tasks such as 'Take the next train', 'Meet someone', and 'Find meeting place'. She accepts the third choice...

Six scenarios Augusto (2007).

Scenario #7:
Based on his context and his expression, the system reveals that Mark would like to spend this weekend for some entertainments. The system recommends him 'Watch movies', 'Play sports', 'Visit friends' based on his habit, context, and preference. Mark chooses the first task. The system recommends him 'Auto-recommend movies' and 'Search movies with constrains'. He selects the first one. The system 'Searches movie titles' which match his preference. After his choice of a movie to watch, the system recommends some tasks such as 'Watch movie at home' and 'Watch movie at movie theatre'. He selects to watch the movie at a theatre. The system 'Searches theatres' which offers the movie at the weekend and meets his preference such as distances and schedules. When he selects one of the recommended list of theatres, the system recommends tasks 'Move by train', 'Move by bus', 'Move by car'. He decides to move by train. The system 'Plans the trips to the theatre' which best suit his schedule, the train schedule, and the movie schedule. When Mark accepts one of the recommended trip plans, the system shows a task 'Invite friends' to join the movie with him. As he accepts the recommendation, the system suggests a list of tasks including 'Contact John', 'Contact Mary', 'Contact Paul'... He decides to contact Mary. Immediately, the system discovers that she is in a meeting and that the meeting is about fifteen minutes to finish. Based on her preference, it recommends him 'Contact Mary in 15 minutes later', 'Send SMS', 'Send email', and 'Send voice message'. He selects to contact her by SMS and the system 'Sends an invitation message' to her... Mark and Mary arrive at the theatre about 20 minutes as usual prior to the movie starting. The system recommends them 'Have coffee', 'Go to book store', and 'Take fresh air'. They choose to have coffee, then the system...

Scenario #8:

Bob are about at a winter early Monday morning. When he enters the kitchen, recognising Bob not having breakfast yet, the system offers two tasks 'Make breakfast' and 'Have a shower'. He chooses the first task. Based on his habit and the contents of the refrigerator, the system suggests some tasks 'Make smoked salmon', 'Make instant coffee', 'Make yoghurt mango', and 'Make soft-boiled egg'. He selects the first two tasks. The system interacts with him to accomplish the tasks according to the availability of devices in the kitchen and the recipes selected. As he is having breakfast and watching TV, the TV 'Synchronises with his personal computer at his office to get his calendar and informs him an appointment' with his colleague at 8am at the Digital Café Restaurant. At the moment, the system recognises the weather getting bad, so it suggests him some next tasks: 'Delay the appointment', 'Change the appointment', 'Go to the restaurant now'. Bob chooses the third task. The system 'Informs his personal driver' taking him to the restaurant. Then, he leaves home while the system 'Turn the lights, the TV, and the heater off' and 'Lock the doors'.

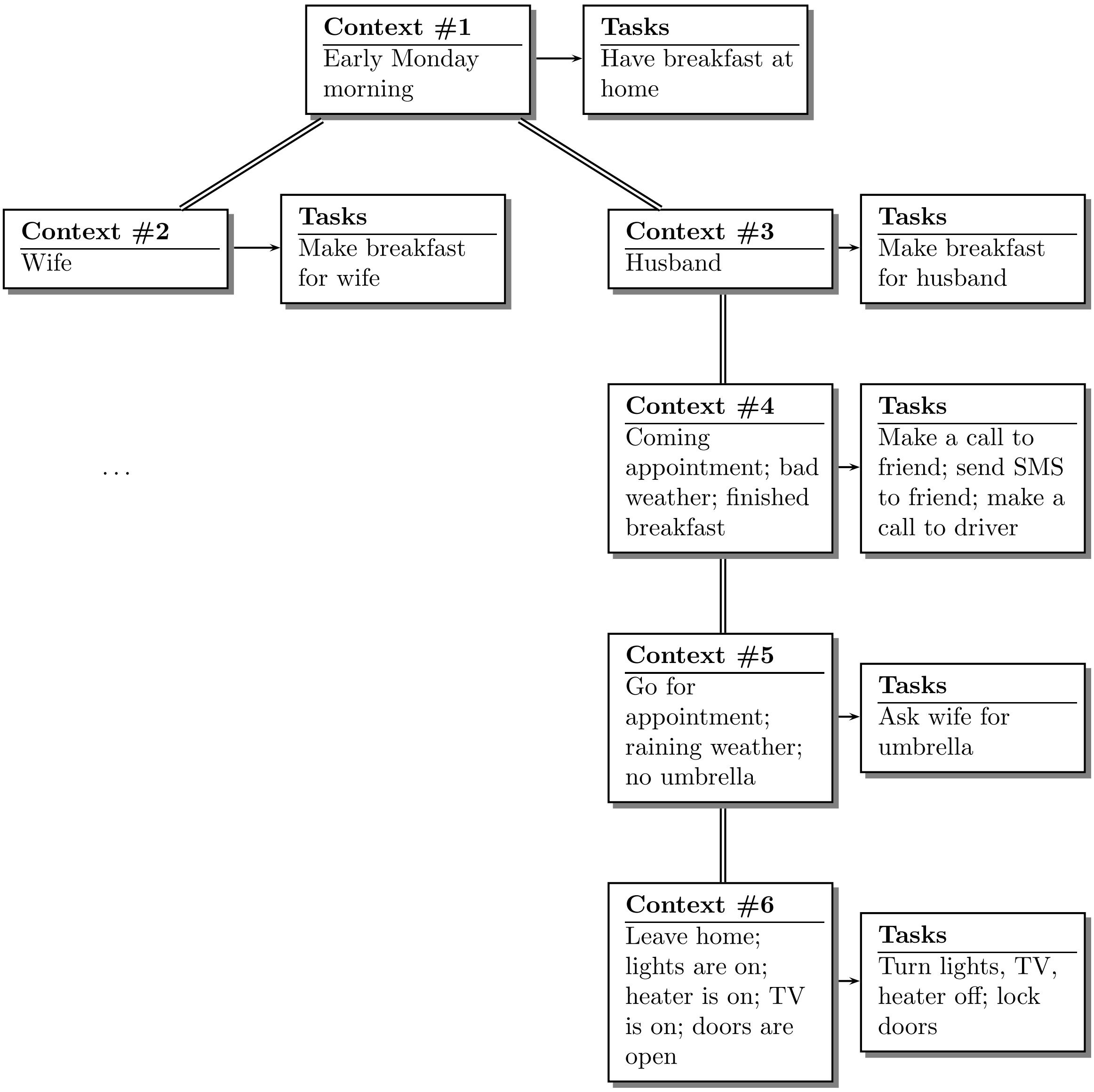

Figure 1: A scenario of context-based tasks selection over the time

3.3 Definitions of Context

Schilit et al. (1994) present the categories of context information as flows:

- *Computing context* includes such as network connectivity, communication costs, communication bandwidth, and nearby resources such as printers, displays, and workstations.
- *User context* includes such as users profile, diary, calendars, feedbacks, roles, preferences, locations, people nearby, activities, users goals, and emotional states.
- *Physical context* includes such as lighting, noise levels, traffic conditions, and weather conditions.
- *Space context* includes such as types of spaces (living rooms, bed rooms, study halls, meeting rooms, squares, stadiums, railway stations, airports), addresses, names, purposes of uses, supported functions.

Chen and Kotz (2000) add *Time context* and *Context history* into the context model by Schilit et al. (1994). Time context includes such as time types (meeting time, lecture time, lunch time, and relaxing time), time of a day, week, month, and season of the year. *Context history* includes such as the previous context which is recorded across a time span.

The authors also classify context information into two groups. Active context which influences the behaviours of an application. An application automatically adapts to discovered context, by changing the application's behaviour such as changing the set of its available features. Passive context is relevant but not critical to an application.

Dey (2001) review previous attempts to define and provide a characterisation of context and context-aware computing. The paper provides an operational definition of context: "*Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*"

Gray and Salber (2001) define sensed context as properties that characterise a phenomenon, are sensed and that are potentially relevant to the tasks supported by an application and/or the means by which those tasks are performed.

Henricksen et al. (2002) examines the nature of context information in pervasive computing environments:

- Context Information Exhibits a Range of Temporal Characteristics;
- Context Information is Imperfect;
- Context Has Many Alternative Representations; and
- Context Information is Highly Interrelated.

Gustavsen (2002); Hofer et al. (2003); Prekop and Burnett (2003) classify context information into external context and internal context. The external context refers to context that can be measured by hardware sensors, e.g., location, light, sound, movement, touch, temperature or air pressure. The internal context is mostly specified by the user or captured by monitoring user interactions, e.g., the user's goals, tasks, work context, business processes, user's emotional states.

Goslar and Schill (2004) outline characteristics of contextual data: *heterogeneity*, *complexity*, *distribution*, *data quality*, *dynamics*, *mutability*, *unavailability*, and *privacy*.

Dourish (2004) argue that context is information (properties) of the real world in which activities are carried out.

Sun and Wu (2005) outline the characteristics of context: *temporality*, *uncertainty*, *concurrency*, and *dependency*.

Coutaz et al. (2005) argue that the view of context-as-process is more flexible than the simpler view of context-as-state. So, they include *current processes* (i.e., what happening at the moment) into the context model.

Reponen and Mihalic (2006) classify context into primary and secondary context. The primary context as the immediate context which is available through human senses, without a need for technical devices to experience it. The secondary context emerges when the content from primary context is sensed by someone via a technical device such as a video camera phone.

Henricksen and Indulska (2006) outline characteristics of context information: heterogeneity in term of quality and persistence, Sensed information is usually highly dynamic and prone to noise and sensing errors, User-supplied information is initially reliable, but easily become out of date, and Imperfectness and ambiguity (or unknowns) of context information when context providers report conflicting context values or fail to report values at all. Authors classify context information into four groups: Sensed, Static, User-supplied, and Derived.

Ni et al. (2006) classify context information into two groups. Physical aspects of user and environment context is environmental contextual information such as time, location, device proximity, lighting condition. Cognitive aspects of user context are such as users' goals, preferences, and emotional state.

Hong et al. (2007) extend the notion of context into three categories: *computing context* (e.g., processors, devices, and bandwidth), *user context* (e.g., user's profile and calendars), and *physical context* (e.g., location, time, lighting, noise levels, traffic conditions, and temperature).

Zimmermann et al. (2007) introduce a context definition comprising three canonical parts: a definition per se in general terms, a formal definition describing the appearance of context and an operational definition characterising the use of context and its dynamic behaviour.

3.3.1 Categories of context information

Computing context includes such as network connectivity, communication costs, communication bandwidth, and nearby resources such as printers, displays, and workstations Schilit et al. (1994).

User context includes such as users profile, diary, calendars, feedbacks, roles, preferences, locations, people nearby, activities, users goals, and emotional states Schilit et al. (1994).

Physical context includes such as lighting, noise levels, traffic conditions, and weather conditions Schilit et al. (1994).

Time context includes such as time types (meeting time, lecture time, lunch time, and relaxing time), time of a day, week, month, and season of the year Chen and Kotz (2000).

Space context includes such as types of spaces (living rooms, bed rooms, study halls, meeting rooms, squares, stadiums, railway stations, airports), addresses, names, purposes of uses, supported functions.

Context history includes such as the previous context which is recorded across a time span Chen and Kotz (2000).

Current processes includes such as what happening at the moment. Coutaz et al. argue that the view of context-as-process is more flexible than the simpler view of context-as-state Coutaz et al. (2005).

3.4 Definitions of Context-Awareness

Schilit et al. (1994) classify context awareness into four categories:

- *Proximation selection/recommendation*: Services, objects, or devices located nearby are emphasised or made easier to use/choose.
- *Automatic contextual reconfiguration*: Functionalities/components of applications are modified according to the context change.
- *Contextual information and commands*: Applications provide/behave information/functions/behaviour based on the current context used.
- *Context-triggered actions*: Applications automatically perform actions according to the discovered context.

Chen and Kotz (2000) classify context awareness into two group: *active context awareness* and *passive context awareness*. Applications of the former automatically and dynamically adapt/tailor their behaviours to response to the discovered context. The later dismiss contextual information or contextual services to interested users.

Dey (2001) defining a context-aware system is "*the system which uses context to provide relevant information and/or services to the user, where relevancy depends on the use's task.*"

3.5 Context Management

Context Toolkit(Dey and Abowd, 1999) and Context Fabric(Hong, 2001) are middleware approaches of context acquisition. They provide benefits: hiding the complexity of the actual sensors used from applications, easy access to context data through querying and notification mechanisms, and reusable and customisable building blocks of context sensing where context data can be used by a variety of applications. Dey et al. (2001)

propose a layered architectural approach (Context Toolkit) with three layers: *context sources*, *context aggregators*, and *context interpreters*.

Chen et al. (2003a); Chen, Finin and Joshi (2004) describe Context Broker Architecture (CoBrA) which uses RDF and OWL for defining and publishing context ontologies, for sharing information about a context, and for reasoning over such information. Central to the architecture is a broker agent that maintains a shared model of the context for all computing entities in the space and enforces the privacy policies defined by the users and devices. Each broker associated with a given space, which can be subdivided into small granularities with individual brokers. This hierarchical approach can avoid the bottlenecks associated with a single centralised broker. The context brokers can infer context knowledge (e.g., user intentions, roles and duties) and can detect and resolve inconsistent knowledge.

Korpipää et al. (2003) present a blackboard based context management framework which permits recognising semantic contexts in real time in the presence of uncertain, noisy, and rapidly changing information and delivering contexts for the terminal applications in an event-based manner. The application programming interface for using semantic context information uses a context ontology to define contexts that clients can use. The framework uses a naive Bayes classifier to recognise higher-level contexts from lower-level context atoms.

Judd and Steenkiste (2003) introduce a Contextual Information Service that provides applications with contextual information via a virtual database. It provides applications with an SQL-like query interface.

Yau et al. (2004) present an adaptive, lightweight and energy-efficient context discovery protocol for ubiquitous environments. This protocol defines the entire process of discovering, acquiring, aggregating and storing contexts.

Liao et al. (2004) propose a Knowledge Management based framework that provides integrated context information about location, weather, time, and user activities to enable context-awareness of m-commerce applications. The framework includes several processes (data collection, preprocessing, integration, modelling, and representation), a context knowledge base, and a knowledge matching engine.

Fahy and Clarke (2004) describe a middleware called CASS which is a database server sitting between context sources and context-aware applications.

Kofod-Peterson and Mikalsen (2005) use subscription mechanisms to notify clients of changes that occur in context. Clients can subscribe to several kinds of changes: (i) changes of the entire current context, (ii) changes in a sub-context in the overall context structure, and (iii) changes in the attribute-value level.

Rajagopalan and Varshney (2006) present a survey of data-aggregation algorithms in wireless sensor networks. Different algorithms are compared and contrasted on the basis of performance measures such as lifetime, latency, and data accuracy.

Devaraju et al. (2007) review approaches to sensory data acquisition and propose a context gathering framework which consists of sensor data model, messaging and communication protocol, and software application programming interface.

Broens et al. (2007) proposes an application infrastructure, coined CACI, which offers a binding transparency between context-aware applications and context producing entities to support rapid development of context-aware applications. CACI enables developers to specify context requirements at a high abstraction level using a declarative binding language. CACI also provides the creation and maintenance of bindings to context producing entities based on such specifications.

Athanasopoulos et al. (2008) present CowsAMI which utilises Web services as interfaces to context sources. Context rules are employed to provide mappings that specify how to populate context relations, with respect to the different context sources that become dynamically available. An underlying context sources discovery mechanism is utilised to maintain context information up to date as context sources, and users get dynamically involved.

Griffin and Pesch (2009) argue that the approach of direct sensor access for context acquisition (Bennett et al., 1994; Laming and Flynn, 1994; Want et al., 1992; **?**) presents a number of problems with regard to extensibility and reusability. It is difficult to extend the application that is gathering the data and because the sensor data acquisition is tightly coupled with the application and the reusability of code is limited.

Kanellopoulos (2009) proposes an ontology-based context management framework that supports context reasoning and context management for adaptive multimedia systems. This model can be applied to each multimedia application and extended by adding new multimedia domain specifications.

Context gathering:

- On-demand(Dey et al., 2001; Hofer et al., 2003).
- On-update(Biegel and Cahill, 2004; Dey et al., 2001; Román et al., 2002).

3.5.1 Architectures to discover context information

Centralised architecture Dey et al. (2001); Lei (2005) A system manages a set of variable names and values of an environment and delivers updates to clients that have previously shown interest by subscribing to the context server. It can also monitor the context changes and send events to interested applications.

Pros Synchronisation, extensibility, rapid development of applications due to encapsulation to separate business logic and graphical user interfaces, hiding low-level sensing details, format of context information is standardised, context sharing support. The costs associated with introducing new context sources can be amortised across many applications, providing a uniform framework for interacting with individual context sources.

Cons Scalability and bottleneck problems.

Distributed architecture Chen and Kotz (2000) A distributed architecture allows context be held at several places. This approach, however, requires the end device to have the capability to sense and process all of the necessary raw contextual information, which may not be efficiently achieved, especially for a complicated trigger and a simple device.

Pros Avoid potential bottleneck problems, no central server is needed, and members can still maintain privacy.

Cons Hard for synchronisation, context sharing issues, various formats of context information, increased computation and communication.

3.5.2 Context query

Garlan et al. Garlan et al. (2002) propose a standard for querying context information. The query has a following form: `QueryResult query(selectedAttribs, serviceName, expression, attribReqs, timeLimit)`, where

`selectedAttribs` – list of attributes selected;
`serviceName` – name of service from which attributes are selected;
`expression` – describes the information desired;
`attribReqs` – requirements such as update time, confidence, etc.;
`timeLimit` – time within which the information should be returned.

3.6 Context Modelling, Representation, and Reasoning

Schmidt et al. (1999) introduce a working model for context:

- A context describes a situation and the environment a device or user is in.
- A context is identified by a unique name.
- For each context, a set of features is relevant.
- For each relevant feature, a range of values is determined (implicit or explicit) by the context.

According to this model, context consists of human factors (e.g., user profiles, social environment, tasks) and physical environment factors (e.g., conditions, infrastructure, location).

Multimodal context Dey et al. (2001).

Gray and Salber (2001) model several important characteristics of sensed context: sensed context types, the subject of sensed context, and properties of sensed context including forms of representation, information quality (e.g., coverage, resolution, accuracy, repeatability, frequency, and timeliness), sensory source (e.g., reliability, intrusiveness, security, cost, and operating profile), data transformations, and actuation.

Henricksen and Indulska (2004, 2006); Henricksen et al. (2002, 2003) describe a context modelling approach (CML) based on Object-Role Modelling (ORM) in which context information is structured around a set of entities, each describing a physical or conceptual object. Properties of entities are represented by attributes. An entity is linked to its attributes and other entities by uni-directional relationships known as associations. Each association originates at a single entity, which is referred to as the owner of the association, and has one or more other participants. Associations can be viewed as assertions about their owning entity, and a context description can correspondingly be viewed as a set of such assertions. In addition to the associations between the entities and their attributes, several associations exist between the entities. The authors provide a graphical notation for the modelling concepts in order to allow context models to be specified diagrammatically. This notation takes the form of a directed graph, in which entity and attribute types form the nodes, and associations are modelled as arcs connecting these nodes.

Held et al. (2002) outline requirements for a model of context: structured, interchangeable, composable/decomposable, uniform, extensible, and standardised. They propose a context model Comprehensive Structured Context Profiles (CSCP) that meets all these requirements. CSCP is based on the Resource Description Framework (RDF). It expresses context information by means of session profiles. A session profile describes all relevant context information of the session. This includes the device profile, the network profile, the user profile, and possibly other context information, such as environmental information.

Network is a graph notation for representing the knowledge of the context in patterns of interacted nodes and links Peters and Shrobe (2003).

Ranganathan and Campbell (2003a) introduce a context model in Gaia.

Prekop and Burnett (2003) present a model of context called Activity-Centric Context. The model focuses on the development of context-aware applications to support complex cognitive activities rather than the user's movement through a physical environment. The key components of the activity-centric view of context are agents and activities. An agent can be a single person, a group of people, or an intelligent machine. The agent is any entity performing the activity. An activity is a description of something being done by the agent. For an activity-centric view of context, context elements would include the resources (e.g., information, computing devices, applications, and other people) needed by the activity and the process for applying them (i.e., how the activity could be performed, and how the resources are applied to achieve the over arching goal of the activity).

Chen et al. (2003b); Chen, Perich, Finin and Joshi (2004) propose context ontology called SOUPA. The ontology is defined using RDF and OWL which provide an explicit semantic representation of context that is suitable for reasoning and knowledge sharing. According to this ontology, the context information is made up of: (1) system users (personal and contact data about system users, such as name, age, address, native language, etc.); (2) user preferences; (3) space information; (4) system services; (5) privacy and security policies that indicate what operations each user can execute; (6) temporal information (date and time, holiday, working day, etc.); (7) services available for a user in the current time; (8) user mobility; (9) user actions (what the user is doing at present moment and what the user has done in the past).

McGrath et al. (2003) introduce ontologies written in DAML+OIL. These ontologies define different kinds of application, service, device, user, data source and other entities. They also describe various relations between the different entities and establish axioms on the properties of these entities. The ontologies also describe different types of contextual information (e.g., locations, activities, weather information, and other information).

Korpipää and Mäntyjärvi (2003) introduce an ontology of context constituents. They refer context as semantic symbolic expressions. Each context constituent is described using seven properties: type, symbolic value, raw value, confidence, source, timestamp, and other additional properties associated with this constituent.

Strang et al. (2003) describe a context model using ontologies called *Aspect-Scale-Context* (ASC) and a *Context Ontology Language* (CoOL). An *aspect* is a classification, symbol- or value-range (e.g., "GeographicCoordinateAspect"), whose subsets are a superset of all reachable states, grouped in one or more related dimensions called *scales* (e.g., "WGS84Scale" and "GaussKruegerScale"). Each *scale* aggregates one or more *context information* (e.g., GaussKruegerCoordinate(367032, 533074)). These core concepts are interrelated via *hasAspect*, *hasScale* and *constructedBy* relations. The ASC model can be used as transfer model to employ the knowledge expressed in other context models. CoOL is grouped into two subsets. *CoOL Core* is a projection into three different ontology languages: OWL, DAML+OIL, and F-Logic. *CoOL Integration* is a collection of schema, protocol, and common sub-concepts enabling the use of *CoOL Core* in several service frameworks.

Tarasewich (2003) presents a "three-dimensional" model of context which consists of three categories of context –environment, participants, and activities– along with any interactions that may exist between and within the categories. Time (e.g., time-of-day, day-of-the-week, months, and seasons) is also incorporated into the model. The "environment" category contains context factors that are outside of the control of the participants. The "participants" category includes the status of the participants. The "activities" category covers participant and activities. "Interactions" deal with those characteristics that pertain to interactions and relationships between individuals, their activities, and the environment.

Wang et al. (2004) outline requirements for a context model: flexible structure, associate to model knowledge, semantic relations, data sharing, and reuse; and having logic reasoning or inference mechanism on raw data to deduce high-level context information.

Strang and Linnhoff-Popien (2004) provide a survey of the approaches to modelling context for ubiquitous computing. Numerous approaches are reviewed, classified, and evaluated with respect to their appropriateness for ubiquitous computing. They outline six requirements for context model approaches: distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality, applicability to existing environments. The approaches are classified by the scheme of data structures: key-value, markup scheme, graphical, object oriented, logic based, ontology based. They conclude that the most promising assets for context modelling is ontology based.

Gu, Wang, Pung and Zhang (2004) and Wang et al. (2004) propose a context model based on ontology (SOCAM) using OWL to address issues including semantic context representation, context reasoning and knowledge sharing, context classification, context dependency, and quality of context. This model supports semantic context representation by defining the common upper ontology for context information in general; and providing a set of low-level ontologies which apply to different sub-domains (e.g., home domain, office domain, vehicle domain, and open space domain). It models the basic concepts of person, location, computational entity, and activity; describes the properties and relationships between these concepts.

Buchholz et al. (2004) introduce a representation language for context information: Comprehensive Structured Context Profiles (CSCP). CSCP is based on the Resource Description Framework (RDF) and thus inherits its interchangeability and extensibility. The authors outline six requirements for a context representation approach: (1) *structured*, (2) *interchangeable*, (3) *decomposable/composable*, (4) *uniform* for all flavours of context data (hardware, user, environment), (5) *extensible*, and (6) *standardised*. Context information is stored in CSCP profiles. A profile describes all context information relevant to a user's session.

Goslar and Schill (2004) provide a model for contextual data using an active semantic network called *Context Map*. The authors outline requirements for a model of contextual information: *formality*, *globality*, *modularity*, *reusability*, *extensibility*, and *history*. The model is based on *topic maps*. The representations of the real world objects in the topic map are called *topics*. These topics are interconnected by *associations* which represent the relations between the real world entities. Topic maps support locality by a concept called *scope*. A scope defines a semantic area in the topic map. Topics associated with a scope are visible only inside its respective area. This allows to distinguish homonymic topics in different contexts.

Preuveneers et al. (2004) propose an OWL based context ontology in CoDAMoS. The ontology is built around four main entities: user, environment, platform, and service.

Ranganathan et al. (2004) propose technique for reasoning about uncertain contexts in pervasive computing environments.

P. Nurmi (2004) define context reasoning as "deducing new information relevant to the use of application(s) and user(s) from the various sources of context-data".

Truong et al. (2005) propose techniques for modelling and reasoning about uncertainty in context-aware systems.

Kofod-Peterson and Mikalsen (2005) divide context into five sub-categories: *environmental context* (e.g., things, services, light, and people); *personal context* (e.g., mood, expertise, disabilities, and weight), *social context* (friends, relatives, and colleagues), *task context* (e.g., what the user is doing, the user's goals, tasks, activities), and *spatio-temporal context* (e.g., time, location, and movement). The authors suggest the use of context templates to create context patterns that constrain context representations and define contextual information in a domain dependent way. Using the context template, application developers define what context structure (contexts with sub contexts), permissible attributes and valid values a context can have.

Strimpakou et al. (2005) develop a context model called DAIDALOS Context Model (DCM) which is based on the object-oriented programming principles. It is built upon the notion of an *Entity* which corresponds to an object of the physical or conceptual world. Each Entity instance is associated with a specific *EntityType* (e.g., person, service, place, terminal, preferences). Entities may demonstrate various *Attributes* (e.g., height, colour, address, location). Each Attribute is related to exactly one Entity, and belongs to a specific *AttributeType*. An Entity may be linked to other Entities via *DirectedAssociations* (DirAs) (e.g., "owns", "uses", "located in", "student of") or *UndirectedAssocations* (UndirAs) (e.g., "friends", "teammates"). The DirAs are relationships among entities with different source and target roles. Each DirA originates at a single entity, called the parent entity, and points to one or more entities, called child entities. The UndirAs do not specify an owner entity, but form generic associations among peer entities. All Entities, Attributes and Associations are marked with a timestamp.

GASChristopoulou and Kameas (2005)

Jie and ZhaoHui (2006) conduct a survey of context reasoning technologies. There are two main techniques: Logic-base (e.g., First Order Logic, Fuzzy logic, Description Logic, and Temporal logic) and learning-based (e.g., Case-based reasoning, Neural networks, and Bayesian approaches).

Meeuwissen et al. (2007) concern inferring and predicting users' context information and user's travelling patterns from time-stamped sequences of location identifiers by using a second-order Markov model and Lempel-Ziv78.

Baldauf et al. (2007) define a context model as the way to express and model context information. The aim is to support for exchanging, reusing, interpreting (reasoning) context information. We can see a context model as a data structure used to express and exchange context information. A context model is also needed to define and store context data in a machine processable form.

Reichle et al. (2008) present requirements for a context model (MUSIC): easy of development, considering the characteristics of mobile and pervasive environments, need for machine-interpretable representation of context information, dealing with special context properties, dealing with context information partitioning, evolution and extensibility, and platform independence, privacy and security issues, support for automatic test execution, logging, simulated operation and visualisation of the system state.

Miraoui et al. (2008b) propose an ontology for modelling contextual information that is enough generic to cover all aspects of context in a pervasive computing system. The ontology is composed of five classes: Device, Sensor, Service, Form, and Context. The relations between these classes will exist whatever the type of device. This permit to affirm that this ontology is enough generic.

Kanellopoulos (2009) describes the ontologies based multimedia context. The context defined is composed of three elements: the user profile, the surroundings, and the communication path (i.e., the elements that characterise the interaction of the users with the communication platform used to access services).

3.6.1 Advantages & Disadvantages of Ontology-Based Models of Context (Agostini et al., 2009; Bettini et al., 2009)

For instance, given two atomic classes Person and Female, the class `Male` can be defined as:
$\mathsf{Male} \equiv \mathsf{Person} \sqcap \neg\mathsf{Female}$.

For another example, the definition of BusinessMeeting:
$\mathsf{BusinessMeeting} \sqsubseteq \mathsf{Activity} \sqcap \geq 2\mathsf{hasActor} \sqcap \forall\mathsf{hasActor.Employee} \sqcap \exists\mathsf{hasLocation.(ConfRoom} \sqcap \mathsf{CompanyBuilding)}$.

Advantages:

- Expressiveness,
- reasoning,
- knowledge sharing.

Disadvantages:

- Very little support for modelling temporal aspects in ontologies.
- The operators provided by OWL-DL are sometimes inadequate to define complex context descriptions (Agostini et al., 2006). For example (Agostini et al., 2009),

consider the isColleagueOf property, a straightforward definition of that property can be given by composing the atomic properties isEmployedBy and isEmployerOf:

isColleagueOf $\equiv$ isEmployedBy $\circ$ isEmployerOf.

Indeed, if a person A is employed by a person B that is the employer of C, then A is colleague of C. Unfortunately, this definition cannot be expressed in OWL-DL. In fact, the language in order to preserve its decidability does not include a constructor for composing relations.

- OWL-DL does not include some expressive class constructors, such as the ones that restrict the membership to a class only to those individual objects that are fillers of two or more properties (these constructors are called *role-value-maps* in the literature). For example, given a property `isCoactorOf` that relates individuals performing an activity together, the role-value-map (`isCoactorOf` $\subseteq$ `isColleagueOf`) defines the class of individuals having as coactors only persons that are their colleagues. If for the sake of simplicity one assumes that an individual cannot perform more than one activity at a time.

- The main problem with reasoning in OWL-DL is computationally expensive. Deriving realisation/instantiation of an individual of interest (e.g., the individual/instance representing the user's activity) in order to find the most specific class the individual belongs to (e.g., a `BusinessMeeting`) through ontological reasoning has *NExpTime* complexity. Hence, online execution of ontological reasoning poses scalability issues, especially when the ontology is populated by a large number of individuals.

Context model is the way to express and model context information. The aim is to support for exchanging, reusing, interpreting (reasoning) context information. We can see a context model as a data structure used to express and exchange context information. A context model is also needed to define and store context data in a machine processable form Baldauf et al. (2007). As mentioned in Wang et al. (2004), there are two requirements of modelling context:

- Flexible structure, associate to model knowledge, semantic relations, data sharing, and reuse;

- Having logic reasoning or inference mechanism on raw data to deduce high-level context information.

Attribute-value model It models the contextual information in key-value pairs.

Tagged encoding or markup scheme model The contexts are modelled as tags and corresponding fields. This model evolved into ConteXtML, which is a simple XMLbased protocol.

Object-oriented model The contextual information is embedded as the states of the object, and the object provides methods to access and modify the states.

Logic-based model expresses contextual information in a domain-centralised database using an entity-relationship data modelMcCarthy and Buvac (1994).

Approaches	Advantages	Disadvantages
Attribute-value model Dey and Abowd (1999); Maass (1998); Schilit et al. (1993); Voelker and Bershad (1994)	• Simple, fast and easy to set and update context attributes • Enable context sharing	• Inflexible and closed, • Insufficient for describing concepts and relationships
Markup scheme model Brown et al. (1997); Indulska et al. (2003)	• Simple, • Flexible, • Structured	• Lack of data ambiguity, • Insufficient for describing formal definitions and relationships
Object-oriented model Cheverst et al. (1999); Hofer et al. (2003); Strimpakou et al. (2005)	• Structured and scalable, • Retrieving high-level concepts, • Solving data ambiguity and consistency	• Inflexible and closed, • Difficult for sharing context
Logic-based model Loke (20-24 Sept. 2004); Ranganathan et al. (2002)	• Allows higher-level context retrieval	• Lack defining structures and relationships, • Lack of data ambiguity
Ontology-based model Bouquet et al. (2003); Chen et al. (2003b); Chen, Perich, Finin and Joshi (2004); Gu, Pung and Zhang (2004); Korpipää et al. (2003); Mokhtar et al. (2006); Preuveneers et al. (2004); Wang et al. (2004)	• Allow defining concepts, entities, properties, and relationships, • Enable context sharing, • Allow context reasoning, • Can be validated with tools as *Jena* or *OWLP*, • Dynamic and flexible, allowing context to be added, deleted, and updated with programming interfaces (*Jena*, *OWL API*), • Many tools support for reasoning on ontologies (*Jena*, *Racer*)	

Table 1: A comparison between approaches to context modelling

Ontology-based model Ontologies represent a description of the concepts and relationships. High and formal expressiveness and the possibilities for applying ontology reasoning techniques make ontologies are the most expressive model and fulfil most of the requirements: simplicity, flexibility and extensibility, generality, expressiveness Strang and Linnhoff-Popien (2004).

There is a notable statement by Baldauf et al. (2007) that currently, there is no standard description language or ontology for sensing contextual information from various sources to enable reuse across various middleware systems and frameworks.

3.7 Context Prediction

WeSigg et al. (2007) propose to base the context prediction procedure on low-level context information. It uses a continuous learning procedure to adapts to possibly changing user habits or environments. A context history is utilised to extract rules describing the typical behaviour of the user. These rules combining with sensor data will be inputs for the context prediction procedure. The task of the context prediction method is to find the context sequence that most likely follows the most recently observed context time series.

3.8 Definitions of Contextual Situations

Situational context (Gellersen et al., 2002) and situation (Dey, 2001; Dobson and Ye, 2006) are the most common ones.

3.9 Situation Modelling and Reasoning

For example, Loke (2006) describes six different ways to specify the situation `In-Meeting-Now` based on contextual cues:

- co-location of people and agenda information;
- co-location of filled coffee cups in a room;
- weight sensors on the floor;
- devices in the room (lights, projector, PowerPoint on PC)
- sounds and noises; and
- cameras ("watch" meeting room for activity).

In each case, the situation `In-Meeting-Now` remains stable and appropriate system actions can simply be associated to this situation, while the contextual cues regarding this situation may change. Additional contextual cues relevant to this situation can be added or obsolete ones can be removed without changing the situation itself, but by modifying only its specification.

3.10 Situation Recognition

One approach to context recognition is for an "expert" to define contexts and user needs in those contexts. Context awareness based on ontologies could be considered as one such approach.

Learning is another possibility for defining contexts, however the approaches are ultimately based on supervised learning which requires the intervention of an expert, or the user, at some point to label contexts or define the user needs in a given context. Furthermore there is still the problem that the learning approach cannot generalise beyond the training data which means either a very large collection of training data or the problem of adapting the learning, through a time dependent gain function, over a long period of time.

Context awareness must be possible to personalise it for each user and their personal mobile device. In both the ontology and supervised learning approaches, personalisation most likely requires intervention by the user which serves to increase rather than decrease the functional complexity of the device. An unsupervised, continuous learning approach to context recognition is now wanted.

3.11 Architectures, Frameworks, Infrastructures, Middlewares, and Toolkits

Kumar and Zambonelli (2007); Mascolo et al. (2002) present surveys of mobile computing middleware.

According to Lei (2005), requirements for frameworks support context-awareness are as follows.

Lightweight restrictions of limited processing power;

Extensibility it is not possible for a single device to sense all context information;

Robustness The architecture has to be robust against disconnections of remote sensors;

Meta-Information The context model has to contain meta-information;

Context-Sharing A mechanism to share the sensed context with other devices.

The Context Toolkit–1999 Dey and Abowd (1999)

Hong and Landay (2001) give a discussion on the differences between kinds of software support for building context-aware applications: libraries, frameworks, toolkits, or infrastructures. Accordingly, a *library* is a generalised set of related algorithms. Libraries focus exclusively on code reuse. On the other hand, *frameworks* concentrate more on design reuse by providing a basic structure for a certain class of applications. Frameworks shoulder the central responsibilities in an application but provide ways to customise the framework for specific needs. *Toolkits* build on frameworks by also offering a large number of reusable components for common functionality. So a graphical user interface event dispatching system would be an example of a framework, and a corresponding toolkit would provide buttons, checkboxes, and text entry fields for that framework. An *infrastructure* is a well-established, pervasive, reliable, and publicly accessible set of technologies that act as a foundation for other systems. An example infrastructure is the Internet itself.

The Sola platform(Chen and Kotz, 2002)

Context Service(Lei et al., 2002)

Gaia project (Román et al., 2002)

Aura(Garlan et al., 2002)

Hofer et al. (2003) proposes a three-layered architecture and a software framework - the Hydrogen Context-Framework - which support context-awareness. It is trimmed to the special requirements of mobile devices regarding particularly the limitations of network connections, limited computing power and the characteristics of mobile users.

An Intelligent Broker for Context-Aware Systems(Chen et al., 2003a)

CARISMA(Capra, 2003)

Hydrogen (Hofer et al., 2003)

MobiPADS(Chuang, 2003)

PICO (Kumar et al., 2003)

CoBrA(Chen et al., 2003b; Chen, Finin, Joshi, Kagal, Perich and Chakraborty, 2004)

CASS(Fahy and Clarke, 2004)

CORTEX(Biegel and Cahill, 2004)

CASS (Fahy and Clarke, 2004)

One.World (Grimm et al., 2004)

ContextPhone(Raento et al., 2005)

SOCAM (Gu, Pung and Zhang, 2004; Gu et al., 2005)

COMPACT(Strimpakou et al., 2006)

A Middleware for Context-Aware Agents in Ubiquitous Computing Environments(Charif and Sabouret, 2006)

Context-Oriented programming(Desmet et al., 2007; Hirschfeld et al., 2008; von Löwis et al., 2007)

Aspect-Oriented Programming(Dantas et al., 2007)

CoWSAMI (Athanasopoulos et al., 2008)

ECORA (Padovitz et al., 2008)

LAICA(Cabri et al., 2005, 2008)

Éamonn Linehan, Tsang and Clarke (2008) present a taxonomy of supporting infrastructures for context-aware computing. This work provides a taxonomic classification of the components of existing context-aware infrastructures. The taxonomy divides the processes involved in supporting context-awareness into three categories: *Adaptation*, *Administration*, and *Gathering*. The *Gathering* category is further decomposed into: *Discovery & Registration*, *Communication*, *Acquisition*, and *Data Formatting*. The *Communication* is further subdivided into three subcategories: *Transport*, *Addressing*, and *Message Protocols*. The *Administration* category is further decomposed into *Aggregation*, *Modelling*, *Persistence*, *Access*, and *Computation Sharing*. *Aggregation* has *Filtering*, *Reasoning*, and *Augmentation*.

Miraoui et al. (2008a) make a survey of architectures that support the development and the implementation of context-aware systems. This survey presents a comparison and evaluation of the architectures on various criteria: context abstraction level, communication model, reasoning system, extensibility and reusability.

System	Architecture	Context model	Resource discovery	Context history	Security & privacy
Context Toolkit	Widget based	Attribute-value tuples	Context interpretation & aggregation	Discoverer component	Context ownership
Gaia	MVC (extended)	$4-ary$ predicates	Context-service module	Discovery service	Secure tracking, location privacy, access control
Hydrogen	Three layered architecture	Object-oriented	Interpretation & aggregation of raw data only	n.a.	n.a.
CoBra	Agent based	Ontologies	Inference engine & knowledge base	n.a.	Rei policy language
Context Management Framework	Blackboard based	Ontologies	Context recognition service	Resource servers & subscription mechanism	n.a.
SOCAM	Distributed with centralised server	Ontologies	Context reasoning engine	Service locating service	n.a.
CASS	Centralised middleware	Relational data model	Inference engine & knowledge base	n.a.	n.a.
CORTEX	Sentient object model	Relational data model	Service discovery framework	Resource management component framework	n.a.

Table 2: A comparison between context-aware systems

3.12 Programming Models & Guidelines

3.12.1 Infrastructure-Based Approaches

Dey and Abowd (1999) define an architecture and present a Java-based Context Toolkit that simplifies context-aware service creation. The toolkit provides three abstract architectural components namely widgets, interpreters and aggregators. These components are responsible for the acquisition of context information from sensors as raw data and the processing of those data to obtain a high-level representation. As a result, this context information can be utilised by context-aware services to achieve their adaptation.

Gray and Salber (2001) outlines a set of activities for design context-aware applications including:

- identifying sensed context possibilities,

- eliciting and assessing information quality requirements,
- eliciting and assessing requirements of the acquisition process,
- consideration of issues of intrusiveness, security, privacy, transformations of the data from source to "consumer", transmission and storage, and
- eliciting and assessing sensor requirements.

Fuentes et al. (2009) propose using the aspect-oriented executable modelling (AOEM) UML profile for designing and simulating pervasive applications. This profile constitutes the basis for debugging these models at design time, before moving into an implementation. Then, these aspect-oriented models are mapped into an aspect-oriented middleware platform for pervasive applications, where the different middleware services are provided as a set of user-configurable aspects. The main benefits of this process are:

- encapsulation of crosscutting concerns, modularisation, reconfigurability, and adaptability,
- systematic model for debugging and testing.

Loke (2009) proposes an incremental approach to building context-aware pervasive systems, with an emphasis on extending over time the contexts and situations a system can be aware of, and creating a formalism in which these systems can be composed. This work presents a formalism of operators for building context-aware pervasive systems incrementally and in a compositional manner. The idea is that a system initially built might only be capable of recognising particular contexts or particular situations of entities, but later, can be extended to recognise more types of context and more situations.

3.12.2 Ad-hoc Solutions

Román et al. (2002) Biegel and Cahill (2004) are examples.

This type of development forces developers to work at a lower level of abstraction by directly programming devices or networks to control them. In addition, if we consider that pervasive systems are characterised by a continuous evolution of hardware and software, the use of ad-hoc solutions makes maintenance and further adaptation extremely difficult E. Serral (2009).

3.12.3 Model Driven Development

Ayed et al. (2007) allows developers to graphically specify context using a UML profile. This approach allows designers to partially obtain the system code for managing context through the definition of modular transformations.

Henricksen and Indulska (2006) propose a graphical context model called CML, which is an extension to Object-Role Modelling for context modelling purposes. The authors also propose a model driven approach to develop context-aware applications based on CML. They propose a semi-automated procedure to map their context models to context management systems that use relational databases.

Ou et al. (2006) attempt to obtain complete functional context-aware pervasive systems by following a MDD strategy. They state that it is necessary to specify the application logic and provide graphical user interfaces for completely developing context-aware pervasive systems. They propose a pure MDA approach for ontology-based context-aware applications development. To do this, they define a set of meta-models and a Model Driven Integration Architecture to integrate these meta-models and generate context-aware application implementations either semi-automatically or automatically. However, one of the drawbacks of this approach is that the abstraction gap is dealt with in only one step, so the transformation can be very complex.

None of the approached above support automatic and complete code generation from models.

E. Serral (2009) introduce a Model Driven Development method for developing context-aware pervasive systems. This method allows to specify a context-aware pervasive system at a high level of abstraction by means of a set of models, which describes both the system functionality and the context information. From these models, an automated code generation strategy is applied to generate the system Java code that provides the system functionality and as well as an OWL specification that represents the context information. Furthermore, this specification is used by a reasoner at runtime to infer context knowledge that is not directly observable, and it is also used by machine learning algorithms to give support to the system adaptation according to the context information.

A. Achilleos (2009) propose a model-driven approach that facilitates the creation of a context modelling framework and simplifies the design and implementation of pervasive services.

3.13 Context-Aware Adaption Techniques

Mozer (1998) develop a system called ACHE that monitors the environment, observes the actions taken by occupants (e.g., adjusting the thermostat; turning on a particular configuration of lights), and attempts to infer patterns in the environment.

Lam Lam and Mostafa (2001) proposes a Bayesian approach to track user interest shift

Gajos et al. (2002) present Alfred – a natural end user programming interface for Intelligent Environments – which allows an end user to “program” the system by telling it the name of a new goal, demonstrating one or more plans for achieving that goal, and finally telling the system the conditions under which it would prefer one plan to another. Upon a user’s request, the system begins recording all of his actions, primarily spoken commands. When the recording is done, he assigns one or more spoken names to the recorded sequence. He can also add hardware triggers to it.

prediction model for navigating category on web by inferencing the user access patterns Chen et al. (2002)

Ranganathan and Campbell (2003b) represent contexts as first-order predicates. The context-aware application developer create a configuration file that associates certain contexts with a method. This method is called when the context becomes true.

Dey et al. (2004) present a CAPpella–a system designed to empower end-users in programming context-aware applications by demonstration. A user of a CAPpella demonstrates a context-aware behaviour that includes both a situation and an associated action to train a CAPpella. Once trained, a CAPpella will enact the demonstrated behaviour: performing the demonstrated action whenever it detects the demonstrated situation. This is like making macro in Microsoft Words.

Korpipää et al. (2004) present Context Studio that allows the user to bind contexts to application actions.

Composite Capabilities/Preference Profiles(CC/PP) Indulska et al. (2003) for describing device capabilities and preferences with a focus on wireless devices such PDAs and mobile phones. It is suitable for expressing very simple preferences, allowing a system to customise preferences. However, future fledged pervasive systems will require much more sophisticated preferences in order to support seamless adaption to changes in these preferences. Thus, this model is not suitable for future pervasive systems.

Kwon and Kim (2004) apply case-based reasoning (CBR). Compared with general AI techniques, CBR systems try to make use of past experiences during problem solving. However, a CBR method in context-aware systems may not work well because it has more opportunities to have a high number of items or variables to consider. That is, as the system becomes more realistic, the number of contexts tends to increase. For example, a system may be implemented using location and weather information. However, as the system increases the sensitivity, more contextual information, such as calendar or emotion, can be newly involved. In that case, the number of criteria will exponentially increase, which affects system feasibility and hence performance. Some supplemental methods to determine weights among the items need to be combined with the CBR method. Kwon and Kim (2004) integrate CBR and multiple decision criteria decision making. The AHP method is adopted as a multiple-criteria decision-making method. The approach imports static preferences from the user preference ontology, and then complete the AHP model by assigning default values to the unspecified contextual preferences. The system then collects cases, and puts any omitted information to have the AHP model work. The final decision is made by comparing the decision from AHP and CBR. If both have the same decision, just do it. Otherwise, the system compares the correctness of past n decisions for the problems. In this case, the decision follows the method which outperforms the other.

Byun and Cheverst (2004) mine rules from context history.

Doctor et al. (2005) propose an unsupervised, data-driven, fuzzy technique for extracting fuzzy membership functions and rules that represent the user's particularised behaviours in an AmI.

identifying similar groups (community) based on their activities and preferences (introduced by Kang and Pisan (2006))

learning the current behavioural patterns of a user (introduced by Kang and Pisan (2006))

Tsang and Clarke (2007) propose mining techniques for selecting the relevant information from a user's behaviour history, for mining usage patterns, and for generating, prioritising, and selecting adaptation behaviour. Their evaluation study shows that the proposed mining approach is more accurate than rule-based and neural network methods when compared to actual user choices.

Anagnostopoulos et al. (2007) present a situation-aware system which enables user to specify action determination rules. A rule associates each situation with a set of certain tasks. When the system infers that the user is involved in a situation, it triggers the tasks associated with the situation.

Rashidi and Cook (2009) introduce CASAS, an adaptive smart home system that utilises machine learning techniques to discover frequent and periodic patterns of daily activities and to generate automation rules that mimic these patterns. CASAS can adapt to the changes in discovered patterns based on the resident implicit and explicit feedback and can automatically update its model to reflect the changes.

Existing context-aware applications commonly adapt to changing context using rules or Machine Learning (ML) techniques. Some also adapt based on historical information.

By End-users:

It offers more acceptance, greater user-centric control, lower deployment costs, better support for personalisation, and frequent upgrade support.

Rule-based systems (sensed situations with associated actions):

Many context-aware systems have been proposed, most of which are rule-based adaptation systems. The rule-based context-aware system allows the user to categorise system's behaviours and to specify their preferences so that the system may build rules for triggering its actions. Rule-based adaptation, however, is very costly in that writing the adaptation rules can be complicated and time-consuming. Hence, the amount of time required to write and maintain a set of rules discourages users from writing rules. The rules are too rigid to flexibly cope with the unspecified conditions, which make the rule-based context-aware system hard to maintain the change of the rules.

Rules must be defined at development time, which means that developers must try to identify "all" possible situations and corresponding actions. Rules are difficult to modify, maintain and scale.

Recognition-based systems/learning-based systems:

To overcome the drawbacks of rule-based systems, a learning-based approach has been regarded as promising as a way to reduce the amount of user efforts to build rules. Algorithms of the learning-based approaches help user in extracting rules from users' past experiences. Learning-based systems do not require the user to do anything, however, an amount of time and patience must be invested while the system gathers sufficiently many examples to learn the user's preferences. Moreover, the learning approach may be slow in adopting the user's quickly changing interests. These lead to the motivation to build a quickly adapting context-aware system.

Probabilistic ML methods such as Bayesian networks are constructed to model context dependencies at development time and so suffer similar maintenance and scalability problems to rule-based approaches. Calculating the necessary probabilities is also a tedious.

The use of neural networks (NN) in context-aware applications has been limited due to their heavy processing requirements and complex black box nature. Existing techniques that store an individuals history are used in conjunction with probabilistic methods and so suffer from the aforementioned problems. Existing techniques that store an individuals history are used in conjunction with probabilistic methods and so suffer from the aforementioned problems. Their accuracy and performance are also affected when historical information is made up of a large number of information types with many possible values. All of the above approaches are also inflexible when adapting to a set of different user goals. Supporting such functionality would require the creation and transition between multiple rule sets, policies, probability tables or networks.

3.14 Prototypes

Standing on user's perspectives, the following categories of context-aware applications are defined by Schilt Schilit et al. (1994).

Proximation selection/recommendation Services, objects, or devices located nearby are emphasised or made easier to use/choose.

Automatic contextual reconfiguration Functionalities/components of applications are modified according to the context change.

Contextual information and commands Applications provide/behave information/functions/b based on the current context used.

Context-triggered actions Applications automatically perform actions according to the discovered context.

Chen et al. Chen and Kotz (2000) based on how context be used in context-aware applications to category them into two classes.

Active context awareness Applications automatically and dynamically adapt/tailor their behaviours to response to the discovered context.

Passive context awareness Applications dismiss contextual information or contextual services to interested users.

Call Forwarding–1992: This application is used to forward calls to the nearest phone of the expected receiver. Locations are context information used by this application.

Teleporting–1994: The system tracks user's location to transfer the applications to the workstation which the user is using.

Mobisaic Web Browser–1994: This web browser appends contextual information such as locations and time into the URLs so that appropriate pages will be retrieved. Whenever the contextual information changes, the web pages will be updated automatically.

Shopping Assistant–1994: The system on user's mobile devices identifies user locations within the store and sends them the information of the products nearby. It also help users to compare prices among products of user's interests.

Active Map–1996: A panel shows a list of users with their locations detected by the system using a wireless base station-based technique.

People and Object Pager–1998: The system determines users's locations and objects' locations to do some things like asking a person nearby a requested object to pick up the object for the requestor or sending a message to a person near by the receiver who has no device with him.

Fieldwork–1998: Context information such as current time and current location is automatically tagged to the records of environmental information of a field when a user is collecting data about the field.

Adaptive GSM phone and PDA-1999: The application adapt devices' properties such as font size according to user's activities and current environmental conditions such as a large font size when the user is walking, otherwise a small font size; a brighter display if light level is low; the profile of the phone is automatically set to an appropriate level based on the recognised context.

PICASSO:–2005 **?**: "PICASSO enables the capture of rich user context (including PC/PDA interactions, audio, video, images, and location) and provides tools that enable users to search, navigate, share, and merge personal events, both from desktop computers and from mobile devices."

3.15 Reminder Applications

Forget-me-not (Laming and Flynn, 1994) is a context-aware reminder employing a small PDA-like device that associates different items of interest with icons to help the person remember various tasks they need to attend to.

Sulawesi (Newman and Clark, 1999) is a spatial reminder service, that uses GPS and infrared to approximate a person's location and delivers reminders accordingly.

MemoClip (Beigl, 2000) is a location-based remembrance appliance. It uses a small wearable computer that relies on location beacons distributed in the environment to trigger location-based reminders.

Cybreminder (Dey and Abowd, 2000) is a reminder application using a variety of context information, including location, to determine when best to trigger reminders.

Dey and Abowd (2000) describe CybreMinder, a prototype context-aware tool that supports users in sending and receiving reminders that can be associated to richly described situations involving time, place, and more sophisticated pieces of context. These situations better define when reminders should be delivered.

Reminder Bracelet (Hansson and Ljungstrand, 2000) is used for notifying subtle cues on mobile devices. It involves a bracelet worn on the wrist of the user that subtly alerts the wearer of upcoming events, as entered into their PDA calendar, using temporal information only.

ComMotion (Marmasse and Schmandt, 2000) is a context-aware reminder system that utilises location as contextual information. Using GPS technology for location-sensing, people could set reminders around certain locations, with given time constraints. When the person was near that location and the timing constraints were satisfied, they would be alerted with an audio alert.

Place-Its (Sohn et al., 2005) is a location-based reminder application that runs on mobile phones. It is used for "placing" a reminder message at a physical location. Reminders are created with a message, and then posted to a location on the person's list of places. They chose to use the location technique employed GSM cell towers.

3.16 Tourist Guides

Stick-e Document (Brown, 1996) Stick-e notes are placed at particular locations using GPS enabled PDAs, and could be made visible to others, thus emulating the affordances of physical notes in a digital environment. Users simply point at a location on the map, the stick-e note(s) for that location then appear.

Cyberguide (Abowd et al., 1997) is a mobile context-aware tour guide which uses knowledge of the user's current location as well as a history of past locations to provide visitors with services concerning location and information. It offers a map of the physical environments that the tourist is visiting; a structured repository of information relating to objects and people of interest in the physical world; information on tourist location and orientation; and a messaging service. Location sensing uses infra-red technology for indoor and GPS for outdoor.

A Stick-e Note Tour of Disney World (Pascoe, 1997) requires the use of a mobile computer, attached to a GPS receiver, to display information on the various attractions as the visitor walks around Disney World. A set of stick-e notes are created which attach the graphics to their associated locations.

Digital Museum (Sakamura, 1998) uses smart cards to detect the proximity of visitors and then provide information about the exhibited objects. The provided information can be based on a static profile stored previously in the smart card.

Hippie (Oppermann et al., 1999; R. and M., 1999) combines the user's location and direction (using Infrared, electronic compass, and GPS) with other information (interests, preferences and knowledge which has been derived from prior interaction) to provide additional details on the exhibits.

HyperAudio (Petrelli et al., 1999) is a mobile guide for PDAs that can give audio descriptions on the exhibit objects depending on the physical location of the users and the amount of time they spend in a certain location as an indication of user interest in certain artworks.

GUIDE (Cheverst et al., 2000, 1999) is a context-sensitive tourist guide for visitors to the city of Lancaster. Visitors are equipped with portable GUIDE units which in turn provide interactive services and dynamically tailored web-based information reflecting the visitor's preferences and environmental context. All information in GUIDE is obtained dynamically using a city-wide wireless network infrastructure.

Spasojevic Rememberer (Fleck et al., 2002) is a tool which offers visitors of museums services for recording their visits. Each record, which can be consulted after the user's visit, consists of a set of web pages with multimedia data, describing the visit. The location of the visitor is identified using infra-red technology and RFID sensors.

Local Location Assistant (Pospischil et al., 2002) is a tourist guide for the city of Vienna, using GPS as localisation technique and a GIS support for generating the maps. It adapts the information to the device, but not to the user's characteristics.

VeGame (Bellotti et al., 2003) is used to explore the city of Venice and learn about its history and architecture through games.

UbiquiTO (Amendola et al., 2004) is a tourist guide for mobile workers in Turin. Services are provided in two different ways: as consequence of explicit request from the user (e.g., to find a hotel or a restaurant); by proactive activation, when the system itself, in specific situations autonomously provides the user with tourist advices. The suggestions are made based on user's interest, proximity, and popularity of items. For location sensing, they use GPS, WLAN, and GSM towers.

COMPASS (van Setten et al., 2004) is a mobile tourist application that adapts its services to the user's needs based on both the user's interests and his current context. In order to provide context-aware recommendations, a recommender system has been integrated with a context-aware application platform.

UbiqMuseum (Cano et al., 2006) provides context-aware information to museum visitors. It uses Bluetooth, WLAN, and Ethernet LAN for communication and service discovery. The system gives information to visitors about what they are viewing, at their level of knowledge, and in their natural language. It can also provide a graphical user interface (GUI) adapted to their device.

m-LOMA (Nurminen, 2006) is designed to be a mobile portal to location-based information in cities. The user can perform textual searches to location-based content, navigate using 3D maps assisted by a GPS, and leave messages to the environment. Its primary focus is on 3D rendering techniques in mobile client.

MyMap (De Carolis et al., 2007) is able to generate personalised presentation of objects of interest starting from an annotated city map. MyMap combines context and user modelling with natural language generation for suggesting to the user what could be interesting to see and do using as interaction metaphor an annotated tourist map.

Personalised Tourist Trip Design Algorithm (Souffriau et al., 2008) applies a combined artificial intelligence and meta-heuristic approach to solve tourist trip design problems. The problem involves a set of possible locations having a score and the objective is to maximise the total score of the visited locations, while keeping the total time (or distance) below the available time budget. The score of a location represents the interest of a tourist in that location. Scores are calculated using the vector space model.

SPETA (Garcia-Crespo et al., 2009) uses knowledge of the user's current location, weather forecast, time, user preferences, friend's recommendation, social networks, as well as a history of past locations in order to provide services that tourists expect from a real tour guide. A GIS system is used.

UbiCicero (Ghiani et al., 2009) is a mobile museum guide equipped with an RFID reader, which detects nearby tagged artworks. It uses the current user position, behaviour history, and the type of device available to provide personalised and relevant information to the user. The information can be displayed on large screens nearby the users.

Location-Aware City Guide (Takeuchi and Sugimoto, 2009) adapts to each user's preferences, and uses an intuitive "metal detector" interface for navigation. The system uses each user's past location data history to estimate individual preferences, and allows users to find shops that match their tastes. A place learning algorithm can find frequented places, and complete with their proper names (e.g., "The Ueno Royal Museum").

3.17 Context-Aware Annotations/Information Triggering/Information Retrieval

Rolling surveys (Brown, 1996) helps field workers gathering data about geographical areas. The field worker divides their area into rectangles, and visits each rectangle and post a stick-e note to record the data. The date of authorship is recorded too. When they next visit a rectangle to re-survey it, the previous note will be displayed and they will prepare a new note to record the latest data.

Audio Aura (Mynatt et al., 1997) and ambientROOM (Ishii and Ullmer, 1997) are systems that play auditory cues for conveying information in the background. These cues can summarise information about the activity of colleagues, notify the status of email or the start of a meeting, and remind of tasks such as retrieving a book at opportune times. They use active badges for location sensing. An active badge is a small electronic tag designed to be worn by a person. It repeatedly emits a unique infrared signal and is detected by a low-cost network of IR sensors placed around a building. A location server combines all the information culled from the IR sensors, perhaps augmenting it with other information such as online calendars and email systems. Audio cues are triggered by changes in the location database and sent to the user's wireless headphones.

Wearable Remembrance Agent (Rhodes, 1997) is a continuously running proactive memory aid that uses the physical context of a wearable computer to provide notes (one-line summaries of notes-files, old email, papers) that might be relevant in that context. These summaries are listed in the bottom few lines of a heads-up display.

Augment-able Reality (Rekimoto et al., 1998) allows users to dynamically attach newly created digital information such as voice notes or photographs to the physical environment, through wearable computers as well as normal computers. Attached data is stored with contextual tags such as location IDs and object IDs that are obtained by wearable sensors, so the same or other wearable users can notice them when they come to the same context.

Graffiti (Burrell and Gay, 2001) allows users to collectively define what's relevant and interesting about a location by posting electronic notes. It could detect user' location and their identity, and would allow the user to receive information related to that context from other users. Users can read notes attached to their current location and can also create notes and attach those notes to specific locations.

GeoNotes (Espinoza et al., 2001) employs Client-Server architecture. Clients can post and retrieve notes associated with places. The sources of positioning information are GPS and GSM.

Hovering information ? is a decentralised, self-organising, infrastructure-free, and location-aware information-dissemination service built over a highly dynamic set of mobile devices. It spreads and disseminates locally produced information among mobile users or applications. It stores individual pieces of hovering information on mobile devices located in the area in which the information was initially produced, even if these devices are unreliable. These pieces act as active entities, detaching themselves from physical media constraints and associating themselves with space and time. They use locally sensed data – such as the direction, position, power, and storage capabilities of nearby mobile devices – to select their next appropriate locations.

The hovering-information service lets applications share (sense and modify) context information related to a precise area and time so that they can adapt their behaviour accordingly.

3.18 Universal Remote Control Applications

Control panel (Brown, 1996) when the user is close to a device such as a copier or FAX machine, she can bring up a control panel on her PDA and control the device.

AIDE (Beigl, 1999) is a generic appliance to interact with devices in the environment. AIDE has a LCD screen to display available commands that can be sent to the controlled device. First, the AIDE device is directed towards the controlled device and the activation key is pressed. A laser beam gives an optical feedback, showing which device is selected. When the controlled device detects the laser beam, it transfers the control description to AIDE using infrared communication. This description containing all possible commands is shown on the AIDE's display. The user selects one of the displayed commands using the keys at the side of AIDE. The selected command is transferred to the controlled device and the action is carried out at the device.

UbiControl (Ringwald, 2002) is a system for using a PDA to control consumer devices found in our environment. By attaching a laser-pointer to the PDA, the user can point to a device in sight and request a user interface description. The user can then control the selected device in a web browser like fashion. A user interface description for every device is stored on the server. After device selection through the laser-pointer, the interface description is downloaded to the PDA. If the user presses a button on the PDA, a request with the encoded command is sent to the server. This command is processed and a result is sent back to the PDA that triggers an update of the displayed interface. The current state of the device is always displayed on the PDA as an indirect feedback to the user action.

Personal Universal Controller (Nichols et al., 2002) provides an intermediary interface (graphical or speech) with which the user interacts as a remote control for any appliance. It downloads a description of the appliance's functions and then automatically creating a high quality interface. The PUC and the appliance exchange messages as the user interacts with the interface.

Uniform (Nichols, Myers and Rothrock, 2006) automatically generates remote control interfaces. It is able to automatically identify similarities between different devices. The similarity information allows the interface generator to use the same type of controls for similar functions, place similar functions so that they can be found with the same navigation steps, and create interfaces that have a similar visual appearance.

Huddle (Nichols, Rothrock, Chau and Myers, 2006) is a system that automatically generates task-based interfaces for a system of multiple appliances based on models of the content flow. Huddle uses knowledge of the appliances' functions and how these functions relate to the content flows to automatically generate a useful set of interfaces, and automatically configure appliances for any set of content flows.

3.19 Mobile Social Computing/Networking Applications

Profile-Based Cooperation (Kortuem et al., 1999) is a way to support informal communication between mobile users during chance encounters. It enables users to publish and exchange personal profile information during physical encounters. It initiates contact between individuals by identifying mutual interests. For example, a diary could keep a record of all individuals we meet during during a conference or trade show ("Tell me who I met today"); during a swap meet, a matchmaker could alert us to the presence of someone who sells a precious item we have been looking for for a long time. Similarly, we could set up our device to advertise items we want to sell so that other people can become aware of us. ("Let me know if someone around here sells a head casket for a 1967 Jaguar E-Type"); a reminder it could alert us to the presence of people we want to meet or talk to in person ("When I meet Howard, remind me that I need to get the key from him").

Hubbub (Isaacs et al., 2002) is a sound-enhanced mobile instant messenger providing awareness information among distributed groups, allowing people to stay connected as they move among multiple fixed locations. It runs on a wireless Palm and a PC, and it makes extensive use of sound as well as visual cues to provide background awareness information without requiring the user's explicit attention.

ActiveClass (Griswold et al., 2004) allows students to ask questions anonymously through a text interface, to answer polls related to the questions, and to give the professor feedback on the class. The students and professor see lists of the questions and polls; students can vote for questions, encouraging the professor to give them precedence.

ActiveCampus Explorer (Griswold et al., 2004) is to let each user see through crowds and undistinguished buildings to reveal nearby friends, potential colleagues, departments, labs, and interesting events. It detects location through the PDA's report of currently sensed 802.11b access points. The reported signal strengths and known locations are used to infer the user's location by a least-squares fit.

WatchMe (Marmasse et al., 2004) is a wristwatch. It is meant to keep intimate friends and family always connected via awareness cues and text, voice instant message, or synchronous voice connectivity. Sensors worn with the watch track location (via GPS), acceleration, and speech activity. When a remote person with whom this information is shared examines it, their face appears on the watch of the person being checked on.

Jabberwocky (Paulos and Goodman, 2004) continuously scans the environment for other Bluetooth devices, and gradually builds a visual map of the familiar strangers that the user encounters.

SmartBlog (Beale, 2005) offers all the regular blogging options for retrieving, categorising, publishing, and editing blog posts on mobile phones. It uses HTTP; therefore, it works over any type of Internet connectivity. The SmartBlog architecture is multithreaded, letting the smart phone function as a mobile phone without compromising its behaviour or performance. No matter SmartBlog's state, the phone should continue to receive and place calls.

Intelligent Multimedia Messaging System (IMMS) (Beale, 2005) places several display units (iPAQs or smart phones) on the office doors of various staff members to act as information and messaging terminals for students. A remote access Web-based management system lets the unit's owner set the display contents, typically a message or image. An SMS-based interface lets users update the display by sending a text message from their phone to the IMMS server. Student members of the department can not only view the image and textual message but also send messages to the owner via the display unit interface. The system also allows remote access to the screen display, so users on the Internet can find lecturers' statuses without having to go to their doors.

BT Share (Beale, 2005) is a peer-to-peer file-sharing system in which a user identifies a file-store on their phone as being public and open for sharing with the group. The system negotiates security protocols and locking mechanisms, letting other authorised users access information in the public store. Users can access, modify, and spread information (documents, music, and so on) among their group without needing a centralised server or explicit communication.

JokeSwap (Beale, 2005) is a joke-sharing application, which lets people exchange jokes over Bluetooth. If someone has a joke in their joke store, the system offers it to other devices. Devices detecting the offer examine their joke stores for the joke, check their owners' personal profiles to see if it's the sort of joke they like, and, if so, accept it and offer a joke in exchange.

Bluedating (Beale, 2005) a localised dating service called Bluedating. Users enter their interests and desires using the interface on a mobile phone, as well as a profile of their desired partner. The system advertises this information (and only this information) over Bluetooth. The application continually searches for other profiles over Bluetooth. When it finds one, it compares the discovered profile to the desired profile. If the two profiles match, the system informs both users (usually by vibrating the phone) of the potential match. The rest is up to the users. This system relies on users' explicit input for updating their profile. Additionally, the broadcast nature of this system leaves room for potential abuse by users. Finally, the system does not guarantee that both users will be aware of the matching.

Telelogs (Davis and Karahalios, 2005) allows profiles in the form of auditory blogs to be shared between familiar strangers. If two people encounter each other more than once, they obtain access to each other's most recent voice blog entry. The information shared depends on the sender of the Telelogs. Crucially, this means that the information could potentially be irrelevant to the recipients. Additionally, users need to record new audio blogs daily in order to keep their profile up to date.

BlueAware (Eagle and , Sandy) runs on mobile devices and scans every 5 minutes for nearby Bluetooth devices. When it detects a new device, it sends the device's BTID to an online server. The server carries out a comparison between the two users' profiles. If there is a match, the server sends both users an alert, along with the photo of the other user, their commonalities, as well as contact information. An issue with this system is the need for establishing communication links with an online server. Also, the recipients of an introductory message are not informed whether or not the other user has been made aware of the receiving message, or if in fact the other person is still in their vicinity.

ContextContacts (Raento et al., 2005) allows for presence and context cues to be shared between users over the network. ContextContacts is used between people who already know each other. Information such as location, time spent there, state of the phone (ringer, vibrator), and number of friends or strangers nearby is shared via servers over the network. This application acts very much like instant messaging applications, and is aimed at enhancing the communication between friends across distances.

Address Book Application (Kostakos et al., 2006) is used to share address books between nearby users. The application runs on mobile devices and allows users securely to exchange the contents of their address books. This exchange reveals only which entries are common to the two users. The application use both Bluetooth and Near Field Communication as an underlying technology.

PeopleTones (Li et al., 2008) is a buddy proximity application for informing users of buddy proximity via peripheral cues from their mobile phones. A sound clip and corresponding vibrotactile pattern is associated with each buddy. For detecting proximity on phones, their algorithm compares cell towers seen by the mobile phone clients.

Clarissa (Gupta et al., 2009) is a location-based mobile social matching application. Clarissa gets the union of known people, social contacts and members of common groups. It then computes a matching score with all the remaining users. This score is computed by assigning higher weights to certain affinity factors (i.e., sports and music). Once the potential matches are identified, Clarissa registers events for the requesters, which are triggered by the co-presence with potential matches during the specified time interval.

Tranzact (Gupta et al., 2009) Its clients send queries for real-time information from various places. For instance, the requester might want to find out the current menu at the cafeteria (which is not posted anywhere outside the cafeteria). In order to answer the query, Tranzact starts by identifying the social contacts of the requester who are currently in the cafeteria. The available contacts receive the request. Responses are sent back.

MoSoSo (Tsai et al., 2009) is a mobile social software based on a P2P network architecture (wireless local area network). The MoSoSo application allows users to discover, communicate and share resources (e.g., files) with one another.

3.20 Context-Aware Phones

Awarenex(Tang et al., 2001) comprises of Contact List and Contact Locator. The Contact List shows whether people are available for contact. The Contact Locator presents more detailed awareness information and the options for contacting that person.

SenSay (Siewiorek et al., 2003) is a context-aware mobile phone, that can manipulate ringer volume, vibration, and phone alerts, provide remote callers with the ability to communicate the urgency of their calls, make call suggestions to users when they are idle, and provide the caller with feedback on the current status of the SenSay user. A number of sensors including accelerometers, light, and microphones are mounted at various points on the body to provide data about the user's context. A decision module uses a set of rules to analyse the sensor data and manage a state machine composed of uninterruptible, idle, active and normal states.

UbiPhone (Hwang et al., 2009) is a human-centred ubiquitous phone system that enables users to make calls by clicking on a callee from a contact list. UbiPhone automatically connects using the most appropriate phone system based on current context information, such as caller and contact's location, presence status, network status, available phone systems, calendars, and social relationships. If contacts aren't available, UbiPhone notifies the caller when they'll be available based on their calendars. If it's an emergency call, UbiPhone provides the caller with the closest person, or knows how to reach them immediately, based on a social network model. UbiPhone also provides an AnyCall service that connects callers to the most appropriate contact from a contact list group.

3.21 Message Filtering and Managing

Nomadic Radio (Sawhney and Schmandt, 2000) is for managing voice and text-based messages in a nomadic environment. Nomadic Radio employs an auditory user interface for navigating among messages as well as asynchronous notification of newly arrived messages. Notification is context sensitive; messages are presented as more or less obtrusive based on importance inferred from content filtering, whether the user is engaged in conversation and his or her own recent responses to prior messages.

3.22 Conference Assistants

Conference Assistant (Dey et al., 1999) is a mobile, context-aware application that assists conference attendees. It uses Context Toolkit as Context Architecture.

EasyMeeting (Chen, Perich, Chakraborty, Finin and Joshi, 2004) provides six services in a meeting room including speech recognition (voice commands), presentation (display PowerPoint presentations on an overhead project), lighting control (adjust the lighting conditions), music (play audio music files), greeting (play a specified greeting message), and display (displaying speaker profiles or reference material on the handheld devices that individual audiences carry). Contexts are the location of meeting participants, the event schedule of a meeting, the presentations that are scheduled for the meeting, the profiles of the presentation speakers, and the state of a meeting.

3.23 Office Assistants

Office Monitor (Yankelovich and McLain, 1996) is a lifelike mannequin placed inside an office to enable office visitors to leave quick messages and to enable office owners to reveal their schedules and location information. The system detects the presence of visitors using a motion sensor and interacts with visitors via a speech interface.

Context-Aware Office Assistants (Yan and Selker, 2000) is an agent that interacts with visitors at the office door and manages the office owner's schedule. The visitor detector consists of two pressure-sensitive mats placed on both side of the office door. The major context information used in the system includes: the identity of the visitor (this is obtained through a question/answer process); the office owner's schedule status (appointment time, appointment content, and available time for appointments); the office owner's busy status (by checking busy tags in calendar data and checking number of people in office).

3.24 Utility Usage Management

4 Related Work

5 Discussion

6 Conclusion

References

A. Achilleos, e. a. (2009). Context modelling and a context-aware framework for pervasive service creation: A model-driven approach, *Pervasive and Mobile Computing* .

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

Abowd, G. D. and Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing, *ACM Trans. Comput.-Hum. Interact.* **7**(1): 29–58.

Agostini, A., Bettini, C. and Riboni, D. (2006). Experience report: Ontological reasoning for context-aware internet services, *Proceedings of the 4th IEEE Conference on Pervasive Computing and Communications Workshops*.

Agostini, A., Bettini, C. and Riboni, D. (2009). Hybrid reasoning in the care middleware for context-awareness, *International Journal of Web Engineering and Technology* **5**(1): 3–23.

Amendola, I., Cena, F., Console, L., Crevola, A., Gena, C., Goy, A., Modeo, S., Perrero, M., Torre, I. and Toso, A. (2004). Ubiquito: A multi-device adaptive guide, *Mobile Human-Computer Interaction – MobileHCI 2004*, pp. 409–414.

Anagnostopoulos, C. B., Ntarladimas, Y. and Hadjiefthymiades, S. (2007). Situational computing: An innovative architecture with imprecise reasoning, *J. Syst. Softw.* **80**(12): 1993–2014.

Athanasopoulos, D., Zarras, A. V., Issarny, V., Pitoura, E. and Vassiliadis, P. (2008). Cowsami: Interface-aware context gathering in ambient intelligence environments, *Pervasive Mob. Comput.* **4**(3): 360–389.

Augusto, J. C. (2007). *Intelligent Computing Everywhere*, Springer London, chapter Ambient Intelligence: The Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence, pp. 213–234.

Ayed, D., Delanote, D. and Berbers, Y. (2007). Mdd approach for the development of context-aware applications, *CONTEXT*, pp. 15–28.

Baldauf, M., Dustdar, S. and Rosenberg, F. (2007). A survey on context-aware systems, *Int. J. Ad Hoc Ubiquitous Comput.* **2**(4): 263–277.

Banavar, G., Beck, J., Gluzberg, E., Munson, J., Sussman, J. and Zukowski, D. (2000). Challenges: an application model for pervasive computing, *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 266–274.

Beale, R. (2005). Supporting social interaction with smart phones, *IEEE Pervasive Computing* **4**(2): 35–41.

Beigl, M. (1999). Point & click - interaction in smart environments, *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, pp. 311–313.

Beigl, M. (2000). Memoclip: A location-based remembrance appliance, *Personal Ubiquitous Comput.* **4**(4): 230–233.

Bellotti, F., Berta, R., De Gloria, A., Ferretti, E. and Margarone, M. (2003). Vegame: Exploring art and history in venice, *Computer* **36**(9): 48–55.

Bennett, F., Richardson, T. and Harter, A. (1994). Teleporting - making applications mobile, *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 82–84.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A. and Riboni, D. (2009). A survey of context modelling and reasoning techniques, *Pervasive and Mobile Computing* **In Press, Corrected Proof**.

Biegel, G. and Cahill, V. (2004). A framework for developing mobile, context-aware applications, *PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, IEEE Computer Society, Washington, DC, USA, p. 361.

Bouquet, P., Giunchiglia, F., Harmelen, F., Serafini, L. and Stuckenschmidt, H. (2003). C-owl: Contextualizing ontologies, *The SemanticWeb - ISWC 2003* pp. 164–179.

Broens, T., van Sinderen, M., van Halteren, A. and Quartel, D. (2007). Dynamic context bindings in pervasive middleware, *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society, Washington, DC, USA, pp. 443–448.

Brown, P., Bovey, J. and Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace, *Personal Communications, IEEE [see also IEEE Wireless Communications]* **4**(5): 58–64.

Brown, P. J. (1996). The stick-e document: a framework for creating context-aware applications, *Proceedings of International Conference on Electronic Documents*.

Buchholz, S., Hamann, T. and Hübsch, G. (2004). Comprehensive structured context profiles (cscp): Design and experiences, *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society, Washington, DC, USA, p. 43.

Burrell, J. and Gay, G. K. (2001). Collectively defining context in a mobile, networked computing environment, *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, pp. 231–232.

Byun, H. and Cheverst, K. (2004). Utilizing context history to provide dynamic adaptations, *Applied Artificial Intelligence* **18**(6): 533–548.

Cabri, G., Ferrari, L., Leonardi, L. and Zambonelli, F. (2005). The laica project: Supporting ambient intelligence via agents and ad-hoc middleware, *Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, IEEE Computer Society, pp. 39–46.

Cabri, G., Mola, F. D., Ferrari, L., Leonardi, L., Quitadamo, R. and Zambonelli, F. (2008). The laica project: An ad-hoc middleware to support ambient intelligence, *Multiagent Grid Syst.* **4**(3): 235–247.

Cano, J.-C., Manzoni, P. and Toh, C.-K. (2006). Ubiqmuseum: A bluetooth and java based context-aware system for ubiquitous computing, *Wireless Personal Communications* **38**(2): 187–202.

Capra, L. (2003). Carisma: Context-aware reflective middleware system for mobile applications, *IEEE Trans. Softw. Eng.* **29**(10): 929–945.

Charif, Y. and Sabouret, N. (2006). An overview of semantic web services composition approaches, *Electr. Notes Theor. Comput. Sci.* **146**(1): 33–41.

Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research, *Technical report*, Dartmouth College, Hanover, NH, USA.

Chen, G. and Kotz, D. (2002). Context aggregation and dissemination in ubiquitous computing systems, *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, p. 105.

Chen, H., Finin, T. and Joshi, A. (2003a). An intelligent broker for context-aware systems, *Adjunct Proceedings of Ubicomp 2003* pp. 183–184. (poster paper).

Chen, H., Finin, T. and Joshi, A. (2003b). An ontology for context-aware pervasive computing environments, *Knowl. Eng. Rev.* **18**(3): 197–207.

Chen, H., Finin, T. and Joshi, A. (2004). A context broker for building smart meeting rooms, *Proc. Knowledge Representation and Ontology for Autonomous Systems Symp.*, AAAI Press, pp. 53–60.

Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F. and Chakraborty, D. (2004). Intelligent agents meet the semantic web in smart spaces, *IEEE Internet Computing* **8**(6): 69–79.

Chen, H., Perich, F., Chakraborty, D., Finin, T. and Joshi, A. (2004). Intelligent agents meet semantic web in a smart meeting room, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, IEEE Computer Society, New York, New York, pp. 854–861.

Chen, H., Perich, F., Finin, T. and Joshi, A. (2004). Soupa: standard ontology for ubiquitous and pervasive applications, *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004* pp. 258–267.

Chen, M., LaPaugh, A. S. and Singh, J. P. (2002). Predicting category accesses for a user in a structured information space, *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, pp. 65–72.

Cheverst, K., Davies, N., Mitchell, K. and Friday, A. (2000). Experiences of developing and deploying a context-aware tourist guide: the guide project, *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, Boston, Massachusetts, United States, pp. 20–31.

Cheverst, K., Mitchell, K. and Davies, N. (1999). Design of an object model for a context sensitive tourist guide, *Computers & Graphics* **23**(6): 883–891.

Christopoulou, E. and Kameas, A. (2005). Gas ontology: an ontology for collaboration among ubiquitous computing devices, *Int. J. Hum.-Comput. Stud.* **62**(5): 664–685.

Chuang, S.-N. (2003). Mobipads: A reflective middleware for context-aware mobile computing, *IEEE Trans. Softw. Eng.* **29**(12): 1072–1085.

Coutaz, J., Crowley, J. L., Dobson, S. and Garlan, D. (2005). Context is key, *Commun. ACM* **48**(3): 49–53.

Dantas, F., Batista, T. and Cacho, N. (2007). Towards aspect-oriented programming for context-aware systems: A comparative study, *Software Engineering for Pervasive Computing Applications, Systems, and Environments, 2007. SEPCASE '07. First International Workshop on*, pp. 4–4.

Davies, N. and Gellersen, H.-W. (2002). Beyond prototypes: Challenges in deploying ubiquitous systems, *IEEE Pervasive Computing* **1**(1): 26–35.

Davis, B. and Karahalios, K. (2005). Telelogs: a social communication space for urban environments, *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, ACM, New York, NY, USA, pp. 231–234.

De Carolis, B., Cozzolongo, G., Pizzutilo, S. and Silvestri, V. (2007). Mymap: Generating personalized tourist descriptions, *Applied Intelligence* **26**(2): 111–124.

Desmet, B., Vallejos, J., Costanza, P. and Hirschfeld, R. (2007). Layered design approach for context-aware systems, *Proceedings of 1st International Workshop on Variability Modelling of Software-intensive Systems (VaMoS 2007)*, Technical Report at Irish Software Engineering Research Centre (Lero), pp. 157–165.

Devaraju, A., Hoh, S. and Hartley, M. (2007). A context gathering framework for context-aware mobile solutions, *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, ACM, Singapore, pp. 39–46.

Dey, A. and Abowd, G. (2000). Cybreminder: A context-aware system for supporting reminders, *Handheld and Ubiquitous Computing*, pp. 201–207.

Dey, A. K. (2001). Understanding and using context, *Personal and Ubiquitous Computing* **5**(1): 4–7.

Dey, A. K. and Abowd, G. D. (1999). The context toolkit: Aiding the development of contextaware applications, *Proceedings of Human Factors in Computing Systems: CHI 99*, ACM Press, Pittsburgh, PA, pp. 434–441.

Dey, A. K., Abowd, G. D. and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, *Human-Computer Interaction* **16**(2): 97–166.

Dey, A. K., Hamid, R., Beckmann, C., Li, I. and Hsu, D. (2004). A CAPpella: Programming by demonstration of context-aware applications, *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, pp. 33–40.

Dey, A. K., Salber, D., Abowd, G. D. and Futakawa, M. (1999). The conference assistant: combining context-awareness with wearable computing, *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pp. 21–28.

Dobson, S. and Ye, J. (2006). Using fibrations for situation identification, *Proceedings of Pervasive 2006 Workshops*.

Doctor, F., Hagras, H. and Callaghan, V. (2005). A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **35**(1): 55–65.

Dourish, P. (2004). What we talk about when we talk about context, *Personal Ubiquitous Comput.* **8**(1): 19–30.

E. Serral, e. a. (2009). Towards the model driven development of context-aware pervasive systems, *Pervasive and Mobile Computing* .

Eagle, N. and (Sandy) Pentland, A. (2006). Reality mining: sensing complex social systems, *Personal Ubiquitous Comput.* **10**(4): 255–268.
ΩÉamonn Linehan et al.

Éamonn Linehan, Tsang, S. L. and Clarke, S. (2008). Supporting context-awareness: A taxonomic review, *Technical Report TCD-CS-2008-37*, Department of Computer Science, Trinity College Dublin.

Edwards, K. K. and Grinter, R. E. (2001). At home with ubiquitous computing: Seven challenges, *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag, London, UK, pp. 256–272.

Edwards, W., Newman, M. and Sedivy, J. (2001). The case for recombinant networking, *Technical report*, Xerox PARC.

Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore, E. and Bylund, M. (2001). Geonotes: Social and navigational aspects of location-based information systems, *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag, London, UK, pp. 2–17.

Estrin, D., Culler, D., Pister, K. and Sukhatme, G. (2002). Connecting the physical world with pervasive networks, *IEEE Pervasive Computing* **1**(1): 59–69.

Fahy, P. and Clarke, S. (2004). Cass - middleware for mobile context-aware applications, *Proceedings of Mobisys 2004 Workshop on Context Awareness*, ACM Press, Boston, Massachusetts, USA, pp. 304–308.

Fleck, M., Frid, M., Kindberg, T., OBrien-Strain, E. and Rajani, M. (2002). Spasojevic rememberer: A tool for capturing museum visits, *Proceedings of the Ubiquitous Computing International Conference, Ubicomp 2002*.

Fox, A., Johanson, B., Hanrahan, P. and Winograd, T. (2000). Integrating information appliances into an interactive workspace, *IEEE Computer Graphics and Applications* **20**(3): 54–65.

Fuentes, L., Gámez, N. and Sánchez, P. (2009). Aspect-oriented design and implementation of context-aware pervasive applications, *Innovations in Systems and Software Engineering* **5**(1): 79–93.

Gajos, K., Fox, H. and Shrobe, H. (2002). End user empowerment in human centered pervasive computing, *Proceedings of the First International Conference on Pervasive Computing (Short paper)*, Zurich, pp. 134–140.

Garcia-Crespo, A., Chamizo, J., Rivera, I., Mencke, M., Colomo-Palacios, R. and Gmez-Berbs, J. M. (2009). Speta: Social pervasive e-tourism advisor, *Telematics and Informatics* **26**(3): 306–315. Mobile and wireless communications: Technologies, applications, business models and diffusion.

Garlan, D., Siewiorek, D., Smailagic, A. and Steenkiste, P. (2002). Project Aura: Toward distraction-free pervasive computing, *IEEE Pervasive Computing* **1**(2): 22–31.

Gellersen, H. W., Schmidt, A. and Beigl, M. (2002). Multi-sensor context-awareness in mobile devices and smart artifacts, *Mobile Networks and Applications* **7**(5): 341–351.

Ghiani, G., Patern, F., Santoro, C. and Spano, L. D. (2009). Ubicicero: A location-aware, multi-device museum guide, *Interacting with Computers* **21**(4): 288 – 303.
URL: *http://www.sciencedirect.com/science/article/B6V0D-4WGBFKG-1/2/36207800e2ab702fabc2d48e9a1121fd*

Goslar, K. and Schill, A. (2004). Modeling contextual information using active data structures, *in* W. Lindner, M. Mesiti, C. Türker, Y. Tzitzikas and A. Vakali (eds), *EDBT Workshops*, Vol. 3268 of *Lecture Notes in Computer Science*, Springer, pp. 325–334.

Gray, P. D. and Salber, D. (2001). Modelling and using sensed context information in the design of interactive applications, *EHCI '01: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, Springer-Verlag, London, UK, pp. 317–336.

Griffin, D. and Pesch, D. (2009). Context in pervasive environments, *Ambient Intelligence with Microsystems* **18**: 187–203.

Grimm, R., Davis, J., Lemar, E., Macbeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S. and Wetherall, D. (2004). System support for pervasive applications, *ACM Trans. Comput. Syst.* **22**(4): 421–486.

Griswold, W. G., Shanahan, P., Brown, S. W., Boyer, R., Ratto, M., Shapiro, R. B. and Truong, T. M. (2004). Activecampus: Experiments in community-oriented ubiquitous computing, *Computer* **37**(10): 73–81.

Gu, T., Pung, H. K. and Zhang, D. Q. (2004). A middleware for building context-aware mobile services, *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th* **5**: 2656–2660.

Gu, T., Pung, H. K. and Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services, *J. Netw. Comput. Appl.* **28**(1): 1–18.

Gu, T., Wang, X., Pung, H. and Zhang, D. (2004). An ontology-based context model in intelligent environments, *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*.

Gupta, A., Kalra, A., Boston, D. and Borcea, C. (2009). Mobisoc: a middleware for mobile social computing applications, *Mob. Netw. Appl.* **14**(1): 35–52.

Gustavsen, R. M. (2002). Condor an application framework for mobility-based context-aware applications, *Proceedings of the Workshop on Concepts and Models for Ubiquitous Computing*.

Hansson, R. and Ljungstrand, P. (2000). The reminder bracelet: subtle notification cues for mobile devices, *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, pp. 323–324.

Held, A., Buchholz, S. and Schill, A. (2002). Modeling of context information for pervasive computing applications, *The 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI2002)*.

Henricksen, K. and Indulska, J. (2004). A software engineering framework for context-aware pervasive computing, *PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, IEEE Computer Society, Washington, DC, USA, p. 77.

Henricksen, K. and Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach, *Pervasive and Mobile Computing* **2**(1): 37–64.

Henricksen, K., Indulska, J. and Rakotonirainy, A. (2001). Infrastructure for pervasive computing: Challenges, *in* K. Bauknecht, W. Brauer and T. Muck (eds), *Informatik*, Osterreichische Computer Gesellschaft, Vienna, Austria, pp. 214–222.

Henricksen, K., Indulska, J. and Rakotonirainy, A. (2002). Modeling context information in pervasive computing systems, *Pervasive '02: Proceedings of the First International Conference on Pervasive Computing*, Springer-Verlag, London, UK, pp. 167–180.

Henricksen, K., Indulska, J. and Rakotonirainy, A. (2003). Generating context management infrastructure from high-level context models, *Fourth International Conference on Mobile Data Management*, Melbourne, Australia, pp. 1–6.

Hirschfeld, R., Costanza, P. and ", O. N. (2008). Context-oriented programming, *Journal of Object Technology* **7**(3): 125–151.

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J. and Retschitzegger, W. (2003). Context-awareness on mobile devices - the hydrogen approach, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9 - Volume 9*, IEEE Computer Society, pp. 292–302.

Hong, D., Chiu, D. K., Shen, V. Y., Cheung, S. C. and Kafeza, E. (2007). Ubiquitous enterprise service adaptations based on contextual user behavior, *Information Systems Frontiers* **9**(4): 343–358.

Hong, J. I. (2001). Context fabric: Infrastructure support for context-aware systems, *Qualifying Exam Proposal* .

Hong, J. I. and Landay, J. A. (2001). An infrastructure approach to context-aware computing, *Hum.-Comput. Interact.* **16**(2): 287–303.

Huang, A. C., Ling, B. C. and Ponnekanti, S. (1999). Pervasive computing: what is it good for?, *MobiDe '99: Proceedings of the 1st ACM international workshop on Data engineering for wireless and mobile access*, ACM, New York, NY, USA, pp. 84–91.

Hwang, R.-H., Tsai, S.-Y. and Wang, C.-Y. (2009). Ubiphone: Human-centered ubiquitous phone system, *Pervasive Computing, IEEE* **8**(2): 40–47.

Indulska, J., Robinson, R., Rakotonirainy, A. and Henricksen, K. (2003). Experiences in using cc/pp in context-aware systems, *Mobile Data Management* pp. 247–261.

Isaacs, E., Walendowski, A. and Ranganthan, D. (2002). Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions, *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 179–186.

Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms, *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 234–241.

Jackson, M. M. (2006). Pervasive direct brain interfaces, *Pervasive Computing* **5**(4): 15–17.

Jazayeri, M. (2002). On the way to pervasive computing (invited paper), *Proceedings of XVI Brazilian Symposium on Software Engineering (SBES'2002), Rio Grande do Sul Brazil*.

Jie, S. and ZhaoHui, W. (2006). Context reasoning technologies in ubiquitous computing environment, *Embedded and Ubiquitous Computing* **4096/2006**: 1027–1036.

Judd, G. and Steenkiste, P. (2003). Providing contextual information to pervasive computing applications, *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, IEEE Computer Society, Washington, DC, USA, pp. 133–142.

Kanellopoulos, D. N. (2009). Adaptive multimedia systems based on intelligent context management, *International Journal of Adaptive and Innovative Systems 2009* **1**(1): 30–43.

Kang, Y.-B. and Pisan, Y. (2006). A survey of major challenges and future directions for next generation pervasive computing, *Proceedings of 21th International Symposium on Computer and Information Sciences - ISCIS 2006*, Vol. 4263 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp. 755–764.

Kofod-Peterson, A. and Mikalsen, M. (2005). Context: Representation and reasoning, representing and reasoning about context in a mobile environment, *Special Issue of the Revue d'Intelligence Artificielle on Applying Context Management* **19**(3): 479–498.

Korpipää, P., Häkkilä, J., Kela, J., Ronkainen, S. and Känsälä, I. (2004). Utilising context ontology in mobile device application personalisation, *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, ACM, New York, NY, USA, pp. 133–140.

Korpipää, P. and Mäntyjärvi, J. (2003). An ontology for mobile device sensor-based context awareness, *Proceedings of 4th International and Interdisciplinary Conference on Modeling and Using Context*, Vol. LNAI 2680, Springer-Verlag, pp. 451–459.

Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H. and Malm, E. (2003). Managing context information in mobile devices, *Pervasive Computing, IEEE* **2**(3): 42–51.

Kortuem, G., Segall, Z. and Thompson, T. G. C. (1999). Close encounters: Supporting mobile collaboration through interchange of user profiles, *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, London, UK, pp. 171–185.

Kostakos, V., O'neill, E. and Shahi, A. (2006). Building common ground for face to face interactions by sharing mobile device context, pp. 222–238.

Kumar, M., Shirazi, B. A., Das, S. K., Sung, B. Y., Levine, D. and Singhal, M. (2003). Pico: A middleware framework for pervasive computing, *IEEE Pervasive Computing* **2**(3): 72–79.

Kumar, M. and Zambonelli, F. (2007). Middleware for pervasive computing, *Pervasive and Mobile Computing* **3**(4): 329–331.

Kwon, O. and Kim, M. (2004). Mymessage: case-based reasoning and multicriteria decision making techniques for intelligent context-aware message filtering, *Expert Syst. Appl.* **27**(3): 467–480.

Lam, W. and Mostafa, J. (2001). Modeling user interest shift using a bayesian approach, *J. Am. Soc. Inf. Sci. Technol.* **52**(5): 416–429.

Laming, M. and Flynn, M. (1994). Forget-me-not: Intimate computing in support of human memory, *Proceedings of FRIEND21 International Symposium Next Generation Human Interfaces*, pp. 125–128.

Lei, H. (2005). Context awareness: a practitioner's perspective, *Ubiquitous Data Management, 2005. UDM 2005. International Workshop on* pp. 43–52.

Lei, H., Sow, D. M., John S. Davis, I., Banavar, G. and Ebling, M. R. (2002). The design and applications of a context service, *SIGMOBILE Mob. Comput. Commun. Rev.* **6**(4): 45–55.

Li, K. A., Sohn, T. Y., Huang, S. and Griswold, W. G. (2008). Peopletones: a system for the detection and notification of buddy proximity on mobile phones, *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, ACM, New York, NY, USA, pp. 160–173.

Liao, S. S., He, J. W. and Tang, T. H. (2004). A framework for context information management, *Journal of Information Science* **30**(6): 528–539.

Loke, S. (2006). On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting, *Proceedings of PerCom Workshops*.

Loke, S. W. (20-24 Sept. 2004). Logic programming for context-aware pervasive computing: Language support, characterizing situations, and integration with the web, *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on* pp. 44–50.

Loke, S. W. (2009). Incremental awareness and compositionality: A design philosophy for context-aware pervasive systems, *Pervasive and Mobile Computing* .

Luther, M., Fukazawa, Y., Wagner, M. and Kurakake, S. (2008). Situational reasoning for task-oriented mobile service recommendation, *The Knowledge Engineering Review* **23**(1): 7–19.

Maass, H. (1998). Location-aware mobile applications based on directory services, *Mob. Netw. Appl.* **3**(2): 157–173.

Marmasse, N. and Schmandt, C. (2000). Location-aware information delivery with commotion, *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, pp. 157–171.

Marmasse, N., Schmandt, C. and Spectre, D. (2004). Watchme: Communication and awareness between members of a closely-knit group, *Ubicomp*, pp. 214–231.

Mascolo, C., Capra, L. and Emmerich, W. (2002). Mobile computing middleware, pp. 20–58.

McCarthy, J. and Buvac, S. (1994). Formalizing context (expanded notes), *Technical report*, Stanford University, Stanford, CA, USA.

McGrath, R., Ranganathan, A., Campbell, R. and Mickunas, D. (2003). Use of ontologies in pervasive computing environments, *Technical Report UIUCDCS-R-2003-2332 UILU-ENG-2003-1719*, Department of Computer Science, University of Illinois at Urbana-Champaign.

Meeuwissen, E., Reinold, P. and Liem, C. (2007). Inferring and predicting context of mobile users, *Bell Lab. Tech. J.* **12**(2): 79–86.

Miraoui, M., Tadj, C. and Amar, C. B. (2008a). Architectural survey of context-aware systems in pervasive computing environment, *Ubiquitous Computing and Communication Journal* **3**(3): 1–9.

Miraoui, M., Tadj, C. and Amar, C. B. (2008b). Context modeling and context-aware service adaptation for pervasive computing systems, *International Journal of Computer and Information Science and Engineering (IJCISE)* **2**(3): 148–157.

Mokhtar, S., Fournier, D., Georgantas, N. and Issarny, V. (2006). Context-aware service composition in pervasive computing environments, *Rapid Integration of Software Engineering Techniques* pp. 129–144.

Mozer, M. C. (1998). The neural network house: An environment that adapts to its inhabitants, *in* M. Coen (ed.), *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, AAAI Press, Menlo Park, CA, pp. 110–114.

Mynatt, E. D., Back, M., Want, R. and Frederick, R. (1997). Audio aura: light-weight audio augmented reality, *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, pp. 211–212.

Newman, N. J. and Clark, A. F. (1999). Sulawesi: A wearable application integration framework, *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, IEEE Computer Society, Washington, DC, USA, p. 170.

Ni, H., Zhou, X., Zhang, D. and Heng, N. (2006). Context-dependent task computing in pervasive environment, *Ubiquitous Computing Systems* pp. 119–128.

Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R. and Pignol, M. (2002). Generating remote control interfaces for complex appliances, *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, pp. 161–170.

Nichols, J., Myers, B. A. and Rothrock, B. (2006). Uniform: automatically generating consistent remote control user interfaces, *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, New York, NY, USA, pp. 611–620.

Nichols, J., Rothrock, B., Chau, D. H. and Myers, B. A. (2006). Huddle: Automatically generating interfaces for systems of multiple connected appliances, *Proceedings of the* 19^{th} *annual ACM symposium on User interface software and technology*, ACM, NY, USA, pp. 279–288.

Nurminen, A. (2006). m-loma - a mobile 3d city map, *Web3D '06: Proceedings of the eleventh international conference on 3D web technology*, ACM, New York, NY, USA, pp. 7–18.

Oppermann, R., Specht, M. and Jaceniak, I. (1999). Hippie: A nomadic information system, *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, Springer-Verlag, London, UK, pp. 330–333.

Ou, S., Georgalas, N., Azmoodeh, M., Yang, K. and Sun, X. (2006). A model driven integration architecture for ontology-based context modelling and context-aware application development, *ECMDA-FA*, pp. 188–197.

P. Nurmi, P. F. (2004). Reasoning in context-aware systems (a possition paper), *Technical report*, Helsinki Institute for Information Technology, Basic Research Unit.

Padovitz, A., Loke, S. and Zaslavsky, A. (2008). Multiple-agent perspectives in reasoning about situations for context-aware pervasive computing systems, *Systems, Man and Cybernetics, Part A, IEEE Transactions on* **38**(4): 729–742.

Pascoe, J. (1997). The stick-e note architecture: extending the interface beyond the user, *Proceedings of the 2nd international conference on Intelligent user interfaces*, ACM, Orlando, Florida, United States, pp. 261–264.

Pascoe, J., Ryan, N. and Morse, D. (1999). Issues in developing context-aware computing, *Handheld and Ubiquitous Computing* pp. 208–221.

Paulos, E. and Goodman, E. (2004). The familiar stranger: anxiety, comfort, and play in public spaces, *CHI '04*, pp. 223–230.

Peters, S. and Shrobe, H. E. (2003). Using semantic networks for knowledge representation in an intelligent environment, *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, IEEE Computer Society, Washington, DC, USA, p. 323.

Petrelli, D., Not, E., Sarini, M., Stock, O., Strapparava, C. and Zancanaro, M. (1999). Hyperaudio: location-awareness + adaptivity, *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, pp. 21–22.

Pospischil, G., Umlauft, M. and Michlmayr, E. (2002). Designing lol@, a mobile tourist guide for umts, *Mobile HCI '02: Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*, Springer-Verlag, London, UK, pp. 140–154.

Prekop, P. and Burnett, M. (2003). Activities, context and ubiquitous computing, *Computer Communications* **26**(11): 1168–1176.

Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V. and De Bosschere, K. (2004). Towards an extensible context ontology for ambient intelligence, *Ambient Intelligence* pp. 148–159.

R., O. and M., S. (1999). A nomadic information system for adaptive exhibition guidance, *Archives and Museum Informatics* **13**: 127–138.

Raento, M., Oulasvirta, A., Petit, R. and Toivonen, H. (2005). Contextphone: a prototyping platform for context-aware mobile applications, *Pervasive Computing, IEEE* **4**(2): 51–59.

Rajagopalan, R. and Varshney, P. (2006). Data-aggregation techniques in sensor networks: a survey, *Communications Surveys & Tutorials, IEEE* **8**(4): 48–63.

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

Ranganathan, A. and Campbell, R. (2003a). A middleware for context-aware agents in ubiquitous computing environments, *Middleware 2003* pp. 998–998.

Ranganathan, A. and Campbell, R. H. (2003b). An infrastructure for context-awareness based on first order logic, *Personal Ubiquitous Comput.* **7**(6): 353–364.

Ranganathan, A., Campbell, R., Ravi, A. and Mahajan, A. (2002). Conchat: a context-aware chat program, *Pervasive Computing, IEEE* **1**(3): 51–57.

Rashidi, P. and Cook, D. J. (2009). Keeping the resident in the loop: Adapting the smart home to the user, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* . to appear.

Reichle, R., Wagner, M., Khan, M., Geihs, K., Lorenzo, J., Valla, M., Fra, C., Paspallis, N. and Papadopoulos, G. (2008). A comprehensive context modeling framework for pervasive computing systems, *Distributed Applications and Interoperable Systems* pp. 281–295.

Rekimoto, J., Ayatsuka, Y. and Hayashi, K. (1998). Augment-able reality: situated communication through physical and digital spaces, *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pp. 68–75.

Reponen, E. and Mihalic, K. (2006). Model of primary and secondary context, *CAI '06: Proceedings of the international workshop in conjunction with AVI 2006 on Context in advanced interfaces*, ACM, New York, NY, USA, pp. 37–38.

Rhodes, B. J. (1997). The wearable remembrance agent: a system for augmented memory, *ISWC '97: Proceedings of the 1st IEEE International Symposium on Wearable Computers*, IEEE Computer Society, Washington, DC, USA, p. 123.

Ringwald, M. (2002). Ubicontrol: providing new and easy ways to interact with various consumer devices, *Adjunct Proceedings of Ubicomp 2002*.

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H. and Nahrstedt, K. (2002). A middleware infrastructure for active spaces, *IEEE Pervasive Computing* **1**(4): 74–83.

Saha, D. and Mukherjee, A. (2003). Pervasive computing: A paradigm for the 21st century, *Computer* **36**(3): 25–31.

Sakamura, K. (1998). Digital museum, *Journal of Information Processing Society of Japan (JIPS)* **39**(5).

Sawhney, N. and Schmandt, C. (2000). Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments, *ACM Trans. Comput.-Hum. Interact.* **7**(3): 353–383.

Schilit, B., Adams, N. and Want, R. (1994). Context-aware computing applications, *Proceedings of Workshop on Mobile Computing Systems and Applications* pp. 85–90.

Schilit, B. N., Theimer, M. M. and Welch, B. B. (1993). Customizing mobile applications, *Mobile & Location-Independent Computing Symposium on Mobile & Location-Independent Computing Symposium*, USENIX Association, Cambridge, Massachusetts, pp. 12–12.

Schmidt, A., Beigl, M. and Gellersen, H.-W. (1999). There is more to context than location, *Computers & Graphics* **23**(6): 893 – 901.

Siewiorek, D., Smailagic, A., Furukawa, J., Krause, A., Moraveji, N., Reiger, K., Shaffer, J. and Wong, F. L. (2003). Sensay: A context-aware mobile phone, *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, IEEE Computer Society, Washington, DC, USA, p. 248.

Sigg, S., Haseloff, S. and David, K. (2007). Prediction of context time series, *in* J. Ikonen, M. Juutilainen and J. Porras (eds), *Proceedings of the 5th Workshop on Applications of Wireless Communications (WAWC'07)*, Lappeenranta, Finland.

Sohn, T., Li, K., Lee, G., Smith, I., Scott, J. and Griswold, W. (2005). Place-its: Location-based reminders on mobile phones, *Proc. Intl. Conf. on Ubiquitous Computing*, pp. 232–250.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Berghe, G. V. and Oudheusden, D. V. (2008). A personalized tourist trip design algorithm for mobile tourist guides, *Applied Artificial Intelligence: An International Journal* **22**(10): 964–985.
URL: *http://www.informaworld.com/10.1080/08839510802379626*

Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey, *First International Workshop on Advanced Context Modeling, Reasoning And Management, UbiComp 2004*.

Strang, T., Linnhoff-Popien, C. and Frank, K. (2003). Cool: A context ontology language to enable contextual interoperability, *Distributed Applications and Interoperable Systems* pp. 236–247.

Strimpakou, M., Roussaki, I., Pils, C., Angermann, M., Robertson, P. and Anagnostou, M. (2005). Context modelling and management in ambient-aware pervasive environments, *in* T. Strang and C. Linnhoff-Popien (eds), *Proceedings of First International Workshop on Location- and Context-Awareness, LoCA 2005*, Vol. 3479 of *Lecture Notes in Computer Science*, Springer, Oberpfaffenhofen, Germany, pp. 2–15.

Strimpakou, M., Roussaki, L., Pils, C. and Anagnostou, M. (2006). Compact: Middleware for context representation and management in pervasive computing, *International Journal of Pervasive Computing and Communications* **2**(3): 229–246.

Sun, J. and Wu, Z. (2005). A comprehensive context model for next generation ubiquitous computing applications, *RTCSA '05: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, IEEE Computer Society, Washington, DC, USA, pp. 447–450.

Takeuchi, Y. and Sugimoto, M. (2009). A user-adaptive city guide system with an unobtrusive navigation interface, *Personal Ubiquitous Comput.* **13**(2): 119–132.

Tang, J. C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F. and Bhalodia, J. (2001). Connexus to awarenex: extending awareness to mobile users, *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, pp. 221–228.

Tarasewich, P. (2003). Towards a comprehensive model of context for mobile and wireless computing, *Proceedings of AMCIS 2003*, pp. 114–124.

Truong, B. A., Lee, Y.-K. and Lee, S.-Y. (2005). Modeling and reasoning about uncertainty in context-aware systems, *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, pp. 102–109.

Tsai, F. S., Han, W., Xu, J. and Chua, H. C. (2009). Design and development of a mobile peer-to-peer social networking application, *Expert Syst. Appl.* **36**(8): 11077–11087.

Tsang, S. L. and Clarke, S. (2007). Mining user models for effective adaptation of context-aware applications, *IPC '07: Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing*, IEEE Computer Society, Washington, DC, USA, pp. 178–187.

van Setten, M., Pokraev, S. and Koolwaaij, J. (2004). Context-aware recommendations in the mobile tourist application compass, *Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, pp. 235–244.

Voelker, G. and Bershad, B. (1994). Mobisaic: an information system for a mobile wireless computing environment, *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pp. 185–190.

von Löwis, M., Denker, M. and Nierstrasz, O. (2007). Context-oriented programming: beyond layers, *Proceedings of the 2007 international conference on Dynamic languages: in conjunction with the 15th International Smalltalk Joint Conference 2007*, ACM, Lugano, Switzerland, pp. 143–156.

Wang, X. H., Zhang, D. Q., Gu, T. and Pung, H. K. (2004). Ontology based context modeling and reasoning using owl, *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society, p. 18.

Want, R., Hopper, A., Falco, V. and Gibbons, J. (1992). The active badge location system, *ACM Trans. Inf. Syst.* **10**(1): 91–102.

Warren, P. W. (2004). From ubiquitous computing to ubiquitous intelligence, *BT Technology Journal* **22**(2): 28–38.

Weiser, M. (1991). The computer for the 21^{st} century, *Scientific American* **3**(265): 94–104.

Weiser, M. (1994). The world is not a desktop, *Interactions* **1**(1): 7–8.

Yan, H. and Selker, T. (2000). Context-aware office assistant, *Proceedings of the 5th international conference on Intelligent user interfaces*, ACM, pp. 276–279.

Yankelovich, N. and McLain, C. D. (1996). Office monitor, *CHI '96: Conference companion on Human factors in computing systems*, ACM, New York, NY, USA, pp. 173–174.

Yau, S. S., Chandraseka, D. and Huang, D. (2004). An adaptive, lightweight and energy-efficient context discovery protocol for ubiquitous computing environments, *Future Trends of Distributed Computing Systems, IEEE International Workshop*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 261–267.

Zimmermann, A., Lorenz, A. and Oppermann, R. (2007). An operational definition of context, *Proceedings of 6th International and Interdisciplinary Conference, CONTEXT 2007*, Roskilde, Denmark, pp. 558–571.