Context for Ubiquitous Data Management

Arthur H. van Bunningen Ling Feng Peter M.G. Apers
Centre for Telematics and Information Technology, Database Group, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
Email: {bunninge,ling,apers}@cs.utwente.nl

Abstract

In response to the advance of ubiquitous computing technologies, we believe that for computer systems to be ubiquitous, they must be context-aware. In this paper, we address the impact of context-awareness on ubiquitous data management. To do this, we overview different characteristics of context in order to develop a clear understanding of context, as well as its implications and requirements for context-aware data management. References to recent research activities and applicable techniques are also provided.

1 Introduction

The research area of ubiquitous computing starts with the vision of Mark Weiser - *to integrate computers in everyday life, ..., having machines that fit human environment instead of forcing humans to enter theirs.* In our view, for computers to be able to fit human environments, they must be in proper size and shape, appropriate for their users, and adaptable to the users' world; in other words, they should be aware of users' context.

Nowadays, context-awareness has sparked vigorous discussions in different fields. However, most current context-aware systems and applications are still small-scaled and use only little context information, such as time, location, and user identity. In the data management area, despite some recent attention to the context-awareness issue, little progress has been made due to the difficulty in capturing, conceptualizing, and representing complicated knowledge about users, context, and tasks [21].

The aim of this paper is to address the impact of context-awareness on ubiquitous data management. We first overview different characteristics of context, as well as its implications and requirements for context-aware data management, from the standpoints of both users and systems. References to recent research activities and applicable techniques are also provided.

The remainder of the paper is organized as follows. Section 2 surveys different definitions of context, context categorization, and context-aware applications. Section 3 describes characteristics of context. Its implications and expectations for context-aware ubiquitous data management are summarized in Section 4. Section 5 concludes the paper.

2 Fundamentals

2.1 Definitions of Context

There are several attempts in the literature to define the notion *context*, ranging from being very broad to being very narrow and application-oriented. In a broad sense, according to Dourish, "*Context is a slippery notion. Perhaps appropriately, it is a concept that keeps to the periphery, and slips away when one attempts to define it*" [20]. Dourish objects against seeing context as something which can be seen as separable from the content of an activity. As an example, he mentions that during a conversation the location of this conversation could turn from context into content when it becomes the subject of this conversation.

Lieberman and Selker look at context from a computer programming's point of view. Traditionally, the field of computer science tries to be *context-independent*: given the same input providing the same output independent of the context of the input [46]. They thus come up with a relatively more concrete definition of context.

Context can be considered to be everything that affects the computation except explicit input and output [46].

Getting close to the application side, one of the most cited definitions of context is probably from Dey *et al.*.

Context is any information that can be used to characterize the situation of an entity. An entity can be a person, place, or object that is considered relevant to the interaction between a user and application, including the user and applications themselves [17].

Proceedings of the 2005 International Workshop on Ubiquitous Data Management (UDM'05)
0-7695-2411-7/05 $20.00 © 2005 IEEE

According to Dey, a system is *context-aware* if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users' task.

Building upon this definition, Gray and Salber clarify the term "interaction" from Dey further by indicating whether it points to what is achieved by doing this interaction (e.g., the task), or the interaction itself (e.g., the user interface or dialogue), and provide a definition for *sensed context* [24].

Sensed context are properties that characterize 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 [24].

Reverting to the data management field, throughout our study [21, 66], we view context as follows.

Context refers to the situation under which user's database access happens [21].

2.2 Context Categorization

There are many possible ways to categorize context information [17, 9, 21, 30]. Here, we describe two kinds of categorization methods, namely, *operational categorization* and *conceptual categorization*. Based on how context is acquired. Henricksen and Indulska categorize context into *sensed, static, profiled,* or *derived* context [30].

Since this categorization is very related to the way context information is acquired, modeled, and treated, we call it **operational categorization**. Contexts of different types differ substantially in how dynamic and reliable they are. In this paper, we will also refer to the *derived context* as *high-level context*, and to the rest as *low-level context*.

Another context categorization is made by Feng *et al.* in [21], which distinguishes *user-centric context* from *environmental context* at a conceptual level. We thus call it **conceptual categorization**.

Most of the context categorizations in the literature fall into either of the two kinds [17, 9].

2.3 Context-Aware Applications

In [40], Korkea-aho provides an overview of existing mobile context-aware applications, which fall into the following four groups, i.e., *office and meeting tools, (tourist) guides, context-aware fieldwork tools,* and *memory aids*. The main context information explored in these systems is user identity, time, and location.

In a further search for "killer applications" for context-awareness, Brown *et al.* classify six kinds of applications; *proactive triggering, streamlining interaction, memory for past events, reminders for future contexts, optimizing patterns of behavior,* and *sharing experiences* [7].

In the domain of information retrieval and service invocation, Dey and Abowd list three possible uses of context in applications: *presentation of information and services to a user, automatic execution of a service,* and *tagging of context to information for later retrieval* [17].

More detailed descriptions of context-aware applications can be found in good surveys [57, 51, 61, 40, 17].

3 Characteristics of Context

In this section, we describe different perspectives of context. The implications of context-awareness for ubiquitous data management will be detailed in the next section.

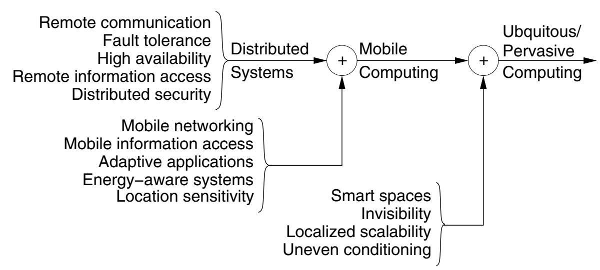

Figure 1. Evolution from distributed computing to ubiquitous/pervasive computing [60]

The characteristics of contextual information are highly influenced and determined by the way it is acquired. Figure 1, reproduced from [60], shows the evolution from *distributed computing*, *mobile computing*, to the current *ubiquitous/pervasive computing* [60, 63]. It is obvious that the acquisition of context will inherently take place among distributed sources in a mobile environment and most characteristics follow from this fact.

3.1 Context is sensed though sensors or sensor networks.

One fundamental characteristic of context is that much context is sensed through sensors or sensor networks [1], for example location or temperature [31, 24]. Data management solutions in this field focus on seeing the sensor network as a database. Some architectural issues, including sensor modeling, imprecise data replication, data compression and prediction, in-network processing, fault tolerance and timeliness, etc. are discussed in [43]. This article focuses on a quality driven approach where a query writer can indicate the confidence s/he wants from an answer (e.g. $\pm 1^{o}C$ of the exact answer). Another system is TinyDB [48], which focuses more on when, where and how the data is acquired; it works on sensors which are running a special operating system (TinyOS) and tries to do as much processing (filtering, aggregation) as possible in the network. An advantage of TinyDB is that it is well documented and available as open source software. A similar database approach to sensor networks is chosen in Cougar [5].

3.2 Context is sensed by small and constrained devices.

What is even more challenging is that sensing of context is done most of the time by cheap, small and (therefore) constrained devices. Cherniack *et al.* point out the limited computing power of such devices, the difficulties to run applications on such a low level and their unreliability [13]. To address another serious consequence of the sensor qualities; the battery capacity, Satyanarayanan goes in more detail about energy costs and energy management, and concludes that energy management has to be done at a high level, like applications switching to modes with lower power consumption when idle [60]. If the switching of operation modes is done by sensors, it will influence the modeling of these sensors [43]. From a data management point of view a related research question is the trade-off of having a sensor based DBMS where on one hand, because of optimizations on sensor level, there is less power spend on transmission, but on the other more power is required by sensors for processing this data.

3.3 Context originates from distributed sources.

As an important aspect mentioned among others in [31, 18, 23], contextual information may come from diverse distributed sources. To get desirable information from these distributed sources, Dey used aggregators to gather context about an entity (e.g. a person) [16]. Sensor querying techniques, such as the one developed in Quasar [43], can also be used to address this issue.

This characteristic brings about the requirement of *high-interrelation* on context-aware data management to be discussed in Section 4, in which we will meanwhile discuss how to integrate the data of these sources.

3.4 Context is continuously changing.

A crucial property of many sorts of context is the continuity, i.e., the user's context constantly changes. This may trigger a system to do new actions, resulting in *proactiveness* [35] but it will also lead to an enormous amount of data to be stored, compressed, and discretized, resulting in impreciseness in the database.

3.5 Context comes from mobile objects.

Closely related to the previous characteristic is the mobility of objects from which to get context information. According to Jones and Brown, mobility is a prime field for context-aware retrieval due to three reasons [35]: Information is now being made available in situations where it was not available before, a mobile user is often in an unfamiliar environment and needs information about that environment, and it is favorable to use context to help to select the information which is needed in this new environment.

Satyanarayanan elaborates two techniques to deal with the mobility of the object and the consequences for information access; *adaptation to the current situation* and *caching* [59].

The mobility aspect of context raises two importance issues, i.e., (moving) spatial/temporal characteristics, and dynamic connections, both to be discussed in Section 4.

3.6 Context has a temporal character.

Because of the mobility, temporal data is very important. Examples for reasoning with time in temporal ontologies for context-awareness are given in [11]. Ter Horst *et al.* introduce the notion of *extended spacetime* to reason about context events which is the set $time \times (space \cup www)$ where $time$ is the set $\Re$ of real numbers, $space$ is the set $\Re^3$, and www is the set of URLs [65]. This way of dealing with knowledge information was introduced by Hayes [29]. Research work on modeling and reasoning with time in Description Logics is detailed in [3].

3.7 Context has a spatial character.

Besides temporariness, the spatial character of context also becomes prominent. In [39], the notion of "activity zones", i.e., regions in which the same activities occur, is proposed to trigger certain events. Harter *et al.* describe a context-aware application which especially focuses on users' location using *Bats* - an ultrasound position determination system [28]. Chen *et al.* also introduce an ontology for both temporal and spatial data [11]. Hightower and Borriello describe techniques of particle filters for location estimation with ultrasound, infrared, and WiFi [33]. A good survey of different techniques for acquiring location information has been done in [32].

3.8 Context information is imperfect and uncertain.

Due to the dynamics, constrained devices, distributed sources, and continuity, etc. there is a high chance that the acquired context information is not perfect. Henricksen and Indulska characterize four types of imperfectness about context information: unknown, ambiguous, imprecise and errorneous[30]

Imperfectness can lead to fuzzy situations where it is for example unclear in which room a person is. Grimm *et al.*

therefore introduce *fuzzy situation descriptions* in ontologies [25]. Both in [26] and [54] a modeling solution for uncertainty is provided by adding a probability predicate and both papers give some references to earlier work. They refer to [19] which describes a simple architecture for incorporation imperfectly sensed context and both use Bayesian networks for reasoning about dependencies between context elements.

Korpipää *et al.* mention uncertainty as well but also combine it with fuzzy situation descriptions, for example *Cold*, *Normal* or *Hot*, and the chances that a situation is like this [42]. According to research by Antifakos *et al.* displaying an indication of the amount of imperfectness of information to a user when this information is used to make decisions will lead to decisions being better [2].

Next to uncertainty about the current context, we are even less sure about the upcoming context [13]. However, we can try to predict behavior by looking at patterns in the behavior, which will require logging.

4 Implications of Context-Awareness

In this section, we discuss the implications and expectations for building context-aware ubiquitous data management systems from the standpoints of both users and systems.

4.1 Users' Perspectives

We focus our discussion on non-functional software requirements. Beyond the so-called "*ilities*" non-functional requirements like reliability, availability, maintainability, responsiveness, manageability, and scalability, etc. [13], we identify four major requirements on context-aware systems from a user's point of view.

4.1.1 Adaptiveness and Personalization

We already saw a growing demand for adaptiveness on mobile, small and constrained devices in mobile computing environments. Adaptiveness and personalization will continue to be a key to context-aware systems. Raghu *et al.* illustrate three methods to achieve personalization [53]:

Rules-based matching Based on user profiles or communities. An example application is "*If the user is a sportsman, display the sport's equipment advertisement.*"

Context-based matching Depending on the current context. An example application is "*If the user is on the sport's page, display the sport's equipment advertisement.*"

Category-based matching Content producers classify their contents based on certain attributes, and users rate their priorities in terms of the same attributes. Also, an agent can steer users to an appropriate content.

4.1.2 Privacy and Security

The mostly mentioned concern for context-awareness throughout our discussions with other researchers is about users' privacy. Also early work on context-awareness done in [52] evidenced that during experiments with tracing users during the day with badges, users did not wear them because of privacy issues.

More research on trust and security concerns of users in ubiquitous computing environments has been reported in [37], which draws a conclusion that other aspects as usability are at least equally important to users. Furthermore, using visible tangible objects to do transactions (e.g. a barcode scanner) can help make transactions more trusted by the users.

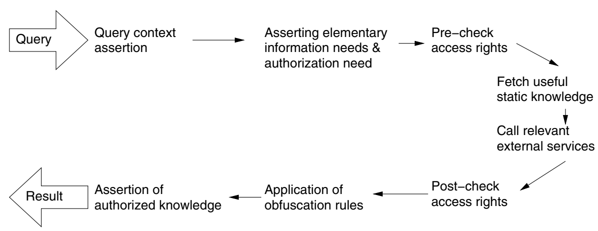

Figure 2. Resolving privacy and security concerns by applying pre- and post-processing [22]

Gandon and Sadeh present in their paper [22] an interesting solution to deal with the privacy and security issue. The context of users is stored in a so called *e-Wallet* and it uses both access rules and obfuscation rules to deliver different context information to different users or applications. Figure 2, taken from [22], shows main steps involved in processing a query submitted to an e-Wallet, while maintaining its user's privacy. In [44] previous study on access rules with an focus on users' location context is reported.

4.1.3 Proactiveness

Proactiveness means to process information on behalf of a user so an action can be taken without requiring his/her attention [35]. This implies knowing what a user would want to do with the requested information, and detecting patterns in her or his behavior.

Proactiveness is viewed as one of the most important requirements for an ubiquitous computing environment [21]. To do this, we need really effective *information extraction* techniques to identify certain situations and some form of *reasoning* mechanisms to determine an appropriate action to take. Tennenhouse even coins the new term *proactive computing* [64] which stands for "*the movement from human-centered to human-supervised (or even unsupervised) computing.*"

4.1.4 Tractability

Tractability means that a user can see why something (proactive) happened. Ideally we would like the proactiveness to be understandable and controllable by users, as suggested in [15]. As Dey *et al.* put it, "*we would ideally handle context the same way as user input.*" [18] Also, from the ubiquitous computing point of view, it could be argued that a human should be able to know what is happening in the background. Dourish even sees displaying the context of a system as one of the few uses of context [20]. In [8] it was mentioned that it should be possible to focus on the tool (the computer) to have it "present-at-hand" in ubiquitous scenarios. This resulted in the following three system design principles:

- Systems should display their own internal states and configuration to the users.
- The deep system structure should be revealed so as to support inspection and adaption.
- Interfaces should offer "direct experience of the structures by which information is organized"

Two remarks should be made; first, some information does not have to be visible all the time but can be made available on request, plus, in many cases its sufficient or even better to give a conceptualization or abstraction of this internal state.

An example here could be the dashboard of a car, by which the user can have the car present-at-hand in case something goes wrong. Another example is the network signal indicator of a mobile phone [8].

4.2 Systems' Perspectives

From the standpoint of systems, context-awareness raises a number of challenges to ubiquitous data management.

4.2.1 Dynamic Connection

Because of the highly-constrained sensors and mobile objects, one serious issue confronting any context-aware system is dynamic connection. That is, connection can be lost when a sensor is out of reach or temporary unavailable, and have to be re-established when it is available again.

In the meantine, data could be cached. On the other hand, observing that information from a not-connected sensor can also be acquired via another sensor or combinations of sensors, Goslar and Schill suggest that a context database should store how to read values and not the current values itself [23]. DeVaul and Pentland present a dynamic decentralized resource discovery framework, which uses semantic descriptions to be able to see what kind of services are available; different components can be registered to a directory registration service when they are available, and deregistered when they are not available anymore [15].

These methods are both very similar to a goal-oriented approach [58, 47]; by having information of what services are available and what they provide, one can combine different available services and abstract from the actual sensors.

A self-organized sensor network approach is also proposed in [50], where autonomous units work together to provide the context related to an object.

Here, it is worth pointing out that the dynamic feature of connections influences the underlying data management strategies. Taking query optimization for example, the techniques developed in [14] are based on the assumption that the network topology changes only slowly, which are therefore not applicable to ubiquitous data management.

4.2.2 Tight Inter-Relationship

Not only does high-level (inferred) context depend on low-level (sensed) context, but also different kinds of low-level context parameters are inter-related. For instance, the amount of computers in a room and the energy usage of this room are closely related.

This tight inter-relation makes it possible to predict some context parameters based on others [31]. Deshpande *et al.* exploit such inter-relations to do optimizations over TinyDB by using correlation between voltage and temperature [14]. However, as noted in [23], because contextual data structures are so highly interconnected, we have to ensure that they are not too complex for limited capabilities of human users and/or local devices. To solve the problem, they suggest breaking the data structures down into smaller parts.

4.2.3 Learning and Reasoning

Due to the inter-relationship among different levels of context, some inference mechanisms are needed in order to derive some context from other contexts. Schmidt is one of the first who did so by using *cues*, which take the value of one sensor and provide a symbolic or subsymbolic output [62]. Taking the output "*the user is running*" and "*the user has a high pulse*" for example, according to several of those cues, a context such as "*the user is jogging*" can be determined. Korpipää *et al.* exploit a set of techniques including Bayesian networks to recognize high-level context, as seen in their procedure in Figure 3 [41]. Combined with the temporal characteristic, research from Höppner could become relevant, who identifies some techniques to discover patterns in time-series [34].

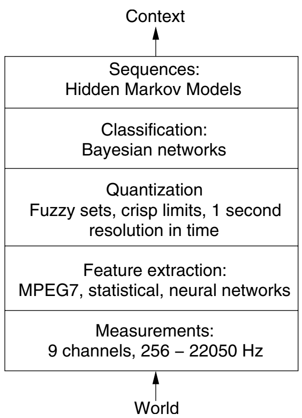

Figure 3. Context derivation layers of [41]

Doing reasoning calls for a way to represent knowledge. An overview of representation languages for context is given in [63]. Some typical languages which have support for reasoning and are nowadays used to describe context are Prolog [56, 55, 12], Clips [22], OWL [67, 27, 10] and First Order Logic [55], [56].

4.2.4 Alternative Representation and Conversion

Confronted with different context information from diverse sensors and possibly from different domains, a flexible context representation mechanism is needed so as to provide conversion among different kinds of context information. In [6] a method of using Prolog rules to convert between different representations is discussed. It is interesting to note that such an alternative context representation problem, in some sense, bears similarity to the schema or data integration problem, which has been extensively addressed in [45] using Description Logics [4].

4.2.5 Metadata about Context Information

Metadata is an effective way to resolve traditional "information overload" problem. It will inevitably play an important role in context-aware ubiquitous computing, which relies heavily on constant context information flow from numerous sensors, monitoring not only the environments, but also users. In [24] an overview of different metadata attributes is given, and divided into the following five main categories: forms of representation, information quality, sensory source, interpretation (data transformation), and actuation (for example, to shut down faulty sensors).

Some other possible metadata includes feature ID, feature value, sensor type ID, sensor location, and time-stamp [50], and accuracy, confidence, update time, and sample interval [36].

Since this is a point where collaboration is necessary between the database side and the sensor side, in our work, we provide a minimum set of requirements for metadata: accuracy information for the measurement, time information for the measurement, possibility of adding "requested accuracy information" to each request, to weight energy cost and accuracy, and possibility of subscribing to a module, so that the sensor will send an update to the subscriber each time a certain event is triggered, and so on.

4.2.6 Storage and Logging of Context Information

Because context-awareness means to be proactive and to detect patterns according to users' behaviors, context information and related reactions thus need to be stored somewhere. Hereby, a number of questions related to what, where, and how to store context information arise [49]. In [49] also a redundant storage approach is discussed, and because of this, it is recommended to store context information at a higher level. This has two other advantages. First, in this way, we can reduce storage space by, only storing the high level context (e.g. being in a meeting), instead of storing all sensor information like temperature and exact location because at this level we can be derive this information. A second advantage is that at a higher level, more computing power is available to do data compression.

5 Conclusions

One major requirement for computer systems to be ubiquitous is to be context-aware. Most current context-aware systems are small scaled and use only little context information. In this paper, we gave an overview of different characteristics of context, its implications and requirements for context-aware computer systems, particularly context-aware data management systems.

We are currently developping and implementing a platform which deals with the implications of context for the data management field using location information on our 140-hectare campus using 650 individual wireless network access points [38].

6 Acknowledgments

This work is funded by the Dutch organization for scientific research (NWO-Vidi project) and the Dutch ministry of economic affairs (MultimediaN and Smart Surroundings project).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magzine*, 40(8):102–114, 2002.

[2] S. Antifakos, A. Schwaninger, and B. Schiele. Evaluating the effects of displaying uncertainty in context-aware applications. In N. Davies, E. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing: 6th International Conference*, pages 54–69. Springer-Verlag Heidelberg, 2004.

[3] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4):171–210, 2000.

[4] F. Baader and W. Nutt. An introduction to description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook*, pages 1–47. Cambridge University Press, 2003.

[5] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In *2nd International Conference on Mobile Data Management*, 2001.

[6] S. Bressan, K. Fynn, C. H. Goh, S. E. Madnick, T. Pena, and M. D. Siegel. Overview of a prolog implementation of the context interchange mediator. In *International Conference and Exhibition on The Practical Applications of Prolog*, 1997.

[7] P. J. Brown, W. Burleson, M. Lamming, O.-W. Rahlff, G. Romano, J. Scholtz, and D. Snowdon. Context- awareness: Some compelling applications. In *CH12000 Workshop on The What, Who, Where, When, Why and How of Context-Awareness*, 2000.

[8] M. Chalmers. A historical view of context. *Computer Supported Cooperative Work (CSCW), The Journal of Collaborative Computing (to appear)*, 2004.

[9] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[10] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18(3):197–207, 2003.

[11] H. Chen, T. Finin, and A. Joshi. Semantic web in a pervasive context-aware architecture. *Artificial Intelligence in Mobile System*, pages 33–40, 2003.

[12] H. Chen, T. Finin, A. Joshi, S. Tolia, and C. Sayers. Creating context-aware software agents. In *First GSFC/JPL Workshop on Radical Agent Concepts*. Springer Verlag (2003), 2002.

[13] M. Cherniack, M. J. Franklin, and S. B. Zdonik. Data management for pervasive computing. Tutorial at VLDB, 2001.

[14] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB 2004*, pages 588–599, 2004.

[15] R. W. DeVaul and A. Pentland. The ektara architecture: The right framework for context-aware wearable and ubiquitous computing applications., 2000.

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

[17] A. K. Dey and G. D. Abowd. Towards a better understanding of context and context-awareness. Technical report GIT-GVU-99-22, Georgia Institute of Technology, 1999.

[18] A. K. Dey, G. D. Abowd, and D. Salber. A context-based infrastructure for smart environments. In *1 st Intl. Workshop on Managing Interactions in Smart Environments (MANSE'99)*, 1999.

[19] A. K. Dey, J. Mankoff, and G. D. Abowd. Distributed mediation of imperfectly sensed context in aware environments. Technical Report GIT-GVU-00-14, Georgia Institute of Technology, 2000.

[20] P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004.

[21] L. Feng, P. M. G. Apers, and W. Jonker. Towards context-aware data management for ambient intelligence. In *Conference on Database and Expert Systems Applications*, 2004.

[22] F. L. Gandon and N. M. Sadeh. Semantic web technologies to reconcile privacy and context awareness. *Web Semantics Journal*, 1(3):241–260, 2004.

[23] K. Goslar and A. Schill. Modeling contextual information using active data structures. In *Workshop for Pervasive Information Management*, 2004.

[24] P. D. Gray and D. Salber. Modelling and using sensed context information in the design of interactive applications. In *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, pages 317–335. Springer-Verlag, 2001.

[25] M. Grimm, M.-R. Tazari, and D. Balfanz. Towards a framework for mobile knowledge management. In *4th international conference on Practical Aspects of Knowledge Management*, 2002.

[26] T. Gu, H. K. Pung, and D. Q. Zhang. A bayesian approach for dealing with uncertain contexts. In *Second International Conference on Pervasive Computing (Pervasive 2004)*, 2004.

[27] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004.

[28] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, pages 59–68. ACM Press, 1999.

[29] P. J. Hayes. Naive physics i: Ontology for liquids. In D. S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 484–502. Kaufmann, San Mateo, CA, 1990.

[30] K. Henricksen and J. Indulska. Modelling and using imperfect context information. In *Second IEEE International Conference on Pervasive Computing and Communications. Workshop on Context Modelling and Reasoning (CoMoRea'04)*, pages 33–37. IEEE Computer Society, 2004.

[31] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In *First International Conference on Pervasive Computing*, pages 167–180, 2002.

[32] J. Hightower and G. Borriello. A survey and taxonomy of location systems for ubiquitous computing. Technical Report UW-CSE 01-08-03, University of Washington, 2001.

[33] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In

N. Davies, E. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing: 6th International Conference*, pages 88–106. Springer-Verlag Heidelberg, 2004.

[34] F. Höppner. *Knowledge Discovery from Sequential Data*. PhD thesis, Technischen Universität Braunschweig, 2003.

[35] G. J. F. Jones and P. J. Brown. Context-aware retrieval for ubiquitous computing environments. In *Mobile HCI Workshop on Mobile and Ubiquitous Information Access*, pages 227–243. Springer, 2004.

[36] G. Judd and P. Steenkiste. Providing contextual information to pervasive computing applications. In *IEEE International Conference on Pervasive Computing (PERCOM)*, pages 133–142, 2003.

[37] T. Kindberg, A. Sellen, and E. Geelhoed. Security and trust in mobile interactions: A study of users perceptions and reasoning. In N. Davies, E. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing: 6th International Conference*, pages 196–213. Springer-Verlag Heidelberg, 2004.

[38] B. Köbben, A. H. van Bunningen, and K. Muthukrishnan. Wireless campus lbs - building campus-wide location based services based on wifi technology. In M. Peterson and E. Stefanakis, editors, *International Workshop on Geographic Hypermedia (in parallel with the Annual Meeting of the Association of American Geographers)*. Springer, 2005.

[39] K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darell. Activity zones for context-aware computing. In *Fifth International Conference on Ubiquitous Computing (UbiComp'03)*, 2003.

[40] M. Korkea-aho. Context-aware applications survey. http://www.hut.fi/∼mkorkeaa/doc/context-aware.html, 2000.

[41] P. Korpipää, M. Koskinen, J. Peltola, S.-M. Mäkelä, and T. Seppänen. Bayesian approach to sensor-based context awareness. *Personal Ubiquitous Computing*, 7(2):113–124, 2003.

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

[43] I. Lazaridis, Q. Han, X. Yu, S. Mehrotra, N. Venkatasubramanian, D. V. Kalashnikov, and W. Yang. Quasar: quality aware sensing architecture. *ACM SIGMOD Record*, 33(1):26–5, 2004.

[44] U. Leonhardt and J. Magee. Security considerations for a distributed location service. *Journal of Network and Systems Management*, 6(1):51–70, 1998.

[45] A. Y. Levy. Logic-based techniques in data integration. In *Logic-based artificial intelligence*, pages 575–595. Kluwer Academic Publishers, 2000.

[46] H. A. Lieberman and T. Selker. Out of context: computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3-4):617–632, 2000.

[47] G. Look and S. Peters. Plan-driven ubiquitous computing. In *Student Oxygen Workshop (SOW'03)*. MIT Project Oxygen, 2003.

[48] S. R. Madden, W. Hong, J. M. Hellerstein, and M. J. Franklin. Tinydb web page. http://telegraph.cs.berkeley.edu/tinydb/, 2004.

[49] B. Meyers and A. Kern. <context-aware> schema </context-aware>. In *CHI Workshop on The What, Who, When, Where, Why, and How of Context-Awareness*, 2000.

[50] F. Michahelles, M. Samulowitz, and B. Schiele. Detecting context in distributed sensor networks by using smart context-aware packets. In *the International Conference on Architecture of Computing Systems: Trends in Network and Pervasive Computing*, pages 34–50. Springer-Verlag, 2002.

[51] K. Mitchell. A survey of context-aware computing. Technical report, Lancaster University, 2002.

[52] W. M. Newman, M. Eldridge, and M. Lamming. Pepys: Generating autobiographies by automatic tracking. In *Second European Conf. on Computer-Supported Cooperative Work*, pages 175–188, 1991.

[53] T. S. Raghu, P. K. Kannan, H. R. Rao, and A. B. Whinston. Dynamic profiling of consumers for customized offerings over the internet: A model and analysis. *Decision Support Systems*, 32(2):117–134, 2001.

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

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

[56] A. Ranganathan, R. H. Campbell, A. Ravi, and A. Mahajan. Conchat: A context-aware chat program. *IEEE Pervasive Computing*, 1(3):52–58, 2002.

[57] K. Rehman. 101 ubiquitous computing applications. http://www-lce.eng.cam.ac.uk/∼kr241/html/101_ubicomp.html, 2001. Date accessed: 6-10-2004.

[58] U. Saif, H. Pham, J. M. Paluska, J. Waterman, C. Terman, and S. Ward. A case for goal-oriented programming semantics. In *System Support for Ubiquitous Computing Workshop at the Fifth Annual Conference on Ubiquitous Computing*, 2004.

[59] M. Satyanarayanan. Accessing information on demand at any location: Mobile information access. *IEEE Personal Communications*, 3(1):26–33, 1996.

[60] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8:10–7, 2001.

[61] B. N. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *the Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.

[62] A. Schmidt. There is more to context than location. *Computers and Graphics Journal*, 23(6):893–901, 1999.

[63] T. Strang. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management associated with the Sixth International Conference on Ubiquitous Computing (UbiComp 2004)*, 2004.

[64] D. Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000.

[65] H. ter Horst, M. van Doorn, N. Kravtsova, W. ten Kate, and D. Siahaan. Context-aware music selection using knowledge on the semantic web. In *Fourteenth Belgium-Netherlands Conference on Artificial Intelligence*, pages 131–138, 2002.

[66] A. H. van Bunningen. Context aware querying - challenges for data management in ambient intelligence. Technical Report TR-CTIT-04-51, University of Twente, P.O. Box 217, 2004.

[67] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *2nd IEEE Conference on Pervasive Computing and Communications, Workshop on Context Modeling and Reasoning*, pages 18–22. IEEE Computer Society, 2004.