Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles

Jakob E. Bardram
Centre for Pervasive Healthcare
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Aarhus N, Denmark
bardram@daimi.au.dk

ABSTRACT

Context-awareness is a key concept in ubiquitous computing, which sometimes seems to be a technology looking for a purpose. In this paper we report on the application of context-aware computing for medical work in hospitals, which has appeared to be a strong case for applying context-aware computing. We present the design of a context-aware pill container and a context-aware hospital bed, both of which reacts and adapts according to what is happening in their context. The applications have been evaluated in a number of workshop with clinicians and patients. Based on this empirical work of designing, developing, and evaluating context-aware clinical applications, the paper outlines some key design principles for a context-awareness framework, supporting the development and deployment of context-aware clinical computer applications.

Keywords

Context-aware computing, ubiquitous computing, Java Context-Aware Framework, JCAF, electronic patient record, context-aware hospital bed

1. INTRODUCTION

Clinicians in modern hospitals are constantly moving around between different work settings, they are engaged in many parallel work activities, and they are constantly engaged in various cooperative problem solving groups [4, 2, 9]. Medical work is accomplished by applying a wide range of documents, schemas, charts, whiteboards, etc., which all are tailored and structured according to the specific work setting in which they are used. For example, the content of whiteboards are adapted to the setting in which they are located, e.g. on a ward or in a surgical clinic. Furthermore, information is often retrieved and used according to a concrete usage situation. During the ward round at the patient's bedside, only the patient's medical record is used, whereas in a radiology conference, x-ray images for all the patients in a ward are displayed on large light-screens. In addition, information is often located where it is mostly used. For example, medical charts are primarily used to pour and document the medication given to patients, and are hence located in the medicine room. In summary, the usage of medical and other information is highly dependent on the concrete usage situation – or in other words, is dependent on the context of use.

Looking at clinical computer systems, including Electronic Patient Records (EPR), these systems often do not incorporate any notion of adaptation to the usage context. None of the EPR systems we have studied have mechanisms for adjusting e.g. a display to specific work situations. The same client user-interface is used in all clinical situations, whether the client is used at a ward, in a surgical clinic, a medicine room, or on a laptop at the patient's bedside. This means that the clinicians need to manually adjust the user-interface to fit the concrete usage setting. For example, the nurse pouring medicine for at number of patients in the medicine room needs to manually find each patient, find the medicine schema, and locate (scroll to) the medicine in question – for all patients. And the physician visiting the patient with a laptop must manually indicate to the EPR who the patient is, and what kind of information is relevant in this ward round context.

The idea of *context-aware computing* was one of the early concepts introduced in some of the pioneering work on ubiquitous computing research [18, 17, 11] and has been subject to extensive research since. 'Context' refers to the physical and social situation in which computational devices are embedded. The goal of context-aware computing is to acquire and utilize information about this context of a device to provide services that are appropriate to the particular setting. For example, a cell phone will always vibrate and newer ring in a concert, if it somehow has knowledge about its current location and the activity going on (i.e. the concert) [16]. In this paper we address the problems of clinical computer systems being unaware about their usage context. In section 2 we will present some examples of context-aware medical applications for use in hospitals. We use these examples to discuss more general design principles for the design, development, deployment, and use of context-aware medical application in section 3. The paper is concluded in section 4.

2. CONTEXT-AWARENESS IN HOSPITALS

In the future we imagine a context-awareness infrastructure in place in a hospital that various clinical applications can access and use to adapt to the context in which they currently are running. To illustrate these ideas, we have been designing and implementing examples of context-aware clinical application, including a context-aware Electronic Patient Record (EPR), a context-aware pill container, and a context-aware hospital bed. In this section we shall consider the usage scenario for these applications and how they are designed.

SAC '04, March 14-17, 2004, Nicosia, Cyprus

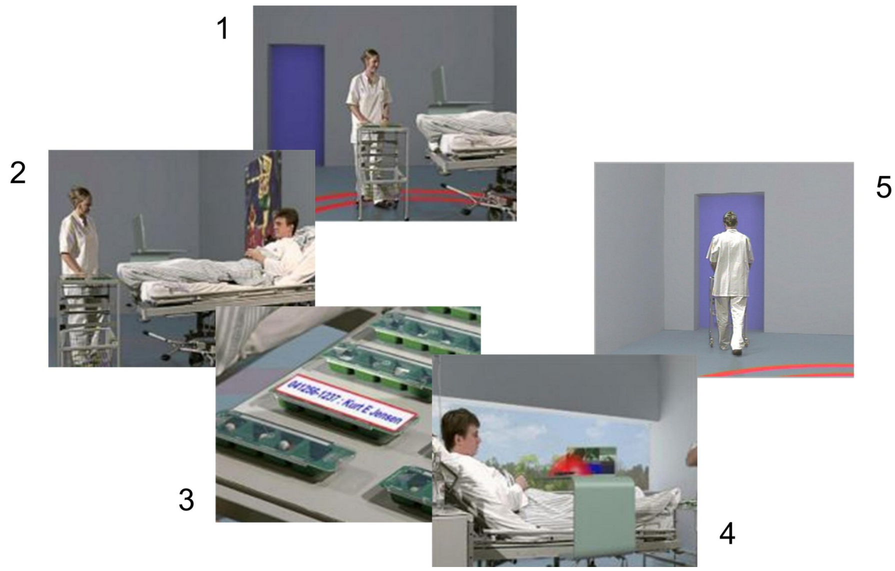

Figure 1: Five scenes from the vision video illustrating context-aware clinical application in a hospital.

2.1 Scenarios of Context-Awareness in Hospitals

The design of context-awareness in hospitals have been illustrated in a video prototype [5]. Figure 1 shows the different scenes in the scenario.

- *Scene 1 – Entering an "Active Zone"*. In this scene we see the nurse entering the ward and walking into the patient's 'zone' as indicated by the circles on the floor. The scene illustrates how a context-awareness infrastructure in the hospital is aware of the location of the nurse; the patient; the trolley; the bed and the computer built into the bed.

- *Scene 2 – The "Context-Aware Hospital Bed"*. This scene illustrates the context-aware hospital bed. This bed has an inbuilt display, which is used by both patients for entertainment purposes (e.g. television) and by the clinicians for accessing medical data while working at the bed. The bed is aware about what happens in its context in the sense that it knows who uses the bed (i.e. the identity of the patient), the patients conditions and treatment, and what and who is near the bed.

- *Scene 3 – The "Context-Aware Pill Container"*. This scene illustrates the use of small, embedded and communicating devices. In this case, the medicine container has computing power, communication capabilities and a display. The patient's medicine container and the patient are aware of each other, enabling the container for the nearby patient to reveal itself. In the video this is done by lighting the proper container with the patient's name.

- *Scene 4 – The "Context-Aware EPR"*. The bed is running a context-aware EPR client and when the patient's pill container is placed on the table, the bed reacts to this change in its environment. Because the bed, via the context-awareness infrastructure, knows the location of the nurse, the patient and the medicine tray, it can tell the computer in the bed to log in the nurse, find the patient's record, display the medicine schema, and in this schema highlight the prescribed medicine, which is in the container.

- *Scene 5 – Leaving the "Active Zone"*. In the last scene on the video we see the nurse leaving the "active zone" of the patient. By doing this the nurse is automatically logged out of the computer and the patient's television program is resumed. The design idea to be illustrated is the easy log on and off by merely approaching computers, a design we have termed 'proximity-based user authentication'.

2.2 Current Implementation

Through an iterative design process in close cooperation with clinicians and patients we have developed a couple of prototype systems, which implement the scenario above. The current versions of the context-aware pill container and hospital bed are illustrated in figure 2 and 3b.

The bed has an integrated computer and a touch sensitive display, and is equipped with various RFID sensors that can identify the patient lying in the bed, the clinician standing beside the bed, and various medical 'stuff' embedding RFID tags. In this way the computer can adapt the computer screen to the users in its vicinity. For example, when the nurse arrives with the patient's medicine, the bed is able to log in the nurse, check if the nurse is carrying the right medicine for the right patient, and it can display the relevant information on the screen, typically the medicine schema from the EPR system. In this case we use the display in the bed to show to the patient (and nurse) that the medicine container is co-located with its patient. We also display a warning if the nurse takes the wrong container. The patient has a profile in the computer system, enabling him or her to read email, browse the internet, or use the screen as a television. The profiles are stored centrally and can hence be migrated across beds in the hospital. The patient is identified using RFID tags in an armband and based on this identification, the personal profile is loaded into the computer.

The design (vision) of the pill container entails departments for medicine, controlled to only reveal the proper portion at a certain time, a finger-print recognition of the patient, and a LED that flashes when the container is close to its patient (see figure 3a). Our current implementation is based on RFID technology for revealing the location and identity of a pill container, e.g. at the bed. Hence, we have not yet implemented mechanisms for flashing a LED on

Figure 2: The Context-Aware Hospital Bed.

the container (figure 3b). When the pill container is detected somewhere close to its patient, we use the context-awareness infrastructure to locate an appropriate screen in the vicinity (because we cannot flash a LED on the container itself). Hence, when the nurse places the container at the patient's bed, we use the display in the bed. Now, because the computer in the bed is running a EPR client, we can also show the corresponding medicine schema.

2.3 Evaluation and Lessons Learned

The design of context-awareness in clinical computer systems in general, and the context-aware pill container and hospital bed in particular, are results of a user-centered design process carried out in cooperation with clinicians and patients. Over the last two years we have carried out 15 workshops, each lasting 4-7 hours, having 5-12 participants of which 8 were clinicians. The participants were asked to role-play a number of scenarios taken from observations of their own work. In some of the workshops, the participant themselves created the scenarios during the workshop. The workshops were video-taped and later analyzed for patterns in use. In the workshops we have evaluated various prototypes ranging from mockups made out of wood and cardboard to running software prototypes. The lessons learned about applying context-awareness to clinical work can be summarized in the following items:

- *Context-awareness is particular useful for user-interface navigation.* During a working day, clinicians need to access a huge amount of medical (and non-medical) data. There is an extensive amount of data associated with the treatment and care of a patient, ranging from medical charts showing biological measurements, medical schemas, x-ray images, medical records, nursing records, etc. In addition, clinicians are involved in the care and treatment of several patients in parallel, which implies a complex juggling and alternation in accessing many different medical records. Actually, a large part of the work and responsibility of medical secretaries is to prepare such medical data in ways that clinicians can access them quickly.

  When clinicians are using the Electronic Patient Records implemented in hospitals, they need to know how to find and access relevant medical data in a specific clinical situation and they must manually navigate to this data. Hence, when a nurse is giving a patient his prescribed medicine, she would need to go to a computer, log in, find the patient, find his medicine schema, scroll to the medicine in question, make a check mark, and log out – all that in order to document that she has given the patient his medicine.

  During the design of systems for context-awareness in clinical work it became obvious that much of this information about e.g. the medication of a patient was available in the usage context of the nurse. Hence, if the computer would understand this context, it could itself navigate to the relevant data – or it might even document it automatically. Using context-awareness for navigating in huge amount of more or less structured medical data turned out to be a strong design and is used in several places in the context-aware EPR system.

- *Context is more than location.* The term context for an entity is in practice often used to denote location of this entity and other entities nearby. For example, the pill container located close to its patient. In a hospital setting, however, context can be a lot more. The nurse documenting medicine above needs not be located close to the patient, even though this patient definitely is part of her current work context. She might be physically located in e.g. the medicine room. Hence, context needs to be more abstract than the location of physical things, and incorporate aspects like patients, their condition, treatment and care, and working schedules for nurses. Furthermore, the *digital context* for a use situation is equally important. For example, how a clinician is using other clinical computer systems, like an EPR, can be valuable context information when trying to deduce what usage context a user is in.

- *Physical things reveal activity.* Work in a hospital involves many physical things. The object of work as such – treatment and care of patients – is not easily digitalized. Hence, a wide range of physical things are used in the daily work at hospitals, e.g. medicine, pill containers, x-ray images, wheelchairs, beds, medicine schemas, charts, medical equipment, specimens, etc. During the design of context-awareness for hospital work it became evident that there was often a simple connection between a clinician accessing some physical things and the activity s/he is engaged in. For example, an x-ray image is typically used by the physician or radiologist for diagnosing the patient. Hence, other relevant data for diagnosing, like medical history, can be shown as well. In general, patient-related 'things' like an x-ray can be used as a physical bookmark used for quickly accessing a patient's medical record. Other things, like the nurse picking up a medicine container, indicate that she is interested in administration of medicine, and the EPR medicine schema can be showed.

- *Using context-awareness to suggest courses of action.* Sensing context information is inherently uncertain and reasoning based on top of this uncertainty can make the whole concept of context-awareness rather vague. For example, when trying to find the relevant medicine data for a patient based on what the context-awareness system thinks is the nurse's current patient might be wrong. Even though the nurse is coming from a patient, it is not certain that it is for this patient she is going to get some medicine. Similarly, it would be rather annoying if a mobile phone frequently turns off its ringing tone because it mistakenly thinks it is at a concert, when it is not (c.f. the example in [16]). Hence, our experience is

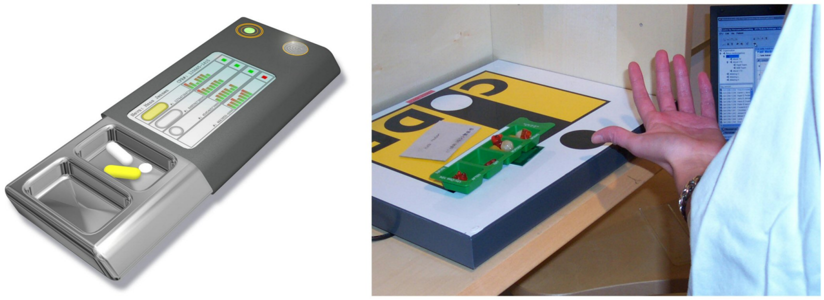

Figure 3: The Context-Aware Pill Container. a – the vision with fingerprint recognition and a LED indicating proximity to the patient. b – the current prototype based on RFID technology.

to avoid having context-aware applications react automatically to (changes in) its context. Rather, suggestions for different courses of action can be presented for the user in a non-intrusive way. In the above example, the context-aware EPR system comes up with two suggestions for the nurse to do; one is to merely show the medicine schema for the last visited patient; the other is to document in the EPR system that the medicine has be given to the patient. The nurse can decide to accept any or none of these suggestions.

3. DESIGN PRINCIPLES FOR CONTEXT-AWARENESS IN MEDICAL WORK

The goal of our research into context-awareness in clinical work is to provide a conceptual and technical framework, which can help application programmers create context-aware clinical computer systems. Such a framework should enable the programmer to design, develop, and deploy application-specific context-awareness features that are required in specific usage settings, while it automatically supports aspects of context-awareness, which are common across applications. This approach is similar to other frameworks and toolkits supporting the development of context-aware application, like the Context Toolkit [10]. Requirements for context-awareness systems and/or frameworks have been widely discussed and described (see e.g. [10, 14, 12, 13, 1]). Therefore, we will only focus on the special requirement we have identified as being central to creating context-aware clinical application in hospitals.

In a hospital there are a wide range of clinical computer systems in use, and new systems are installed and removed on a regular basis. Furthermore, many clinicians (typical research active doctors) build their own applications, such as quality databases supporting a specific clinical experiment. In order to make such applications context-aware there is a need for a stable infrastructure that can be accessed by these applications, and there is a need for a programming interface used by the developers of such applications. The basic design principle in a context-awareness framework for hospitals is therefore to divide it into two parts. One part supports the deployment of 'context services', which are robust, scalable, flexible, adaptable, extensible, etc. Such services run independently of the applications supplying or using context information. The other part enables developers of context-aware applications to represent, acquire, handle, store, and use context information. We refer to the first part as a *Context-awareness Runtime Infrastructure* and to the latter as a *Context-awareness Programming Framework* (or Application Programmer Interface (API)). Let us consider the design principles for these two parts of our framework.

3.1 Runtime Infrastructure

A basic context-awareness infrastructure should serve applications entering and leaving a hospital, enabling them to connect to context-awareness services in their proximity. Hence, the context-awareness infrastructure should run independent of the applications, which on the other hand should know how to access these context-awareness services. This principle of having an independent runtime infrastructure for context-awareness is similar to the Trivial Context System (TCoS) [14] and the Owl Context Service [15]. Let us consider some of the basic design principles for this infrastructure[1]

- *Distributed and Cooperating Services* – Gathering and applying context information is often tied to specific spaces or environments dedicated to a specific purpose. For example, using a context-aware computer system to aid and guide a surgeon is highly dependent on accurate and detailed context information about things going on in the operating room. However, such detailed information is often not particular interesting for activities taking place in the rest of the hospital. Hence, context services may be dedicated to a specific purpose, like handling context information in an operating room. Therefore, a context-awareness infrastructure should be distributed and loosely coupled, while maintaining ways of cooperating in a peer-to-peer fashion. This is in contrast to the TCoS and the Owl systems, which use a centralized context server.

- *Security and Privacy* – Clinical data about patients are important context data for clinical applications, and such data should be handled secure and its privacy respected. For example, the hospital bed uses information about the treatment of the patient as context information, enabling it to adjust itself to the patient. Hence, context data should be protected, subject to access control, and not revealed to unauthorized

[1]There are some basic software architecture qualities for any kind of infrastructure, including this context-awareness infrastructure (see e.g. [6]). These include performance, security, availability, scalability, extensibility, modifiability, portability, integrability, and others. Clearly, this infrastructure should address most of these qualities also, but this is out of scope of this paper. Here we focus only on the design principles concerning context-awareness in hospitals.

clients. Therefore, the context services should embed an access control mechanism. Furthermore, it is important to know the validity of clients delivering context data. A central question is, whether context services can trust the data delivered from a client. This requires an authentication mechanism for clients, and a secure communication link between clients and services. Similar requirements apply for the communication between services.

- *Lookup and Discovery* – Context-aware clinical application will continuously enter and leave the hospital, e.g. running on mobile equipment or being deployed as new applications. Such clients should be able to locate and connect to relevant context services in the infrastructure. Services are therefore required to register at Lookup and Discovery services and reveal what they can do. This lookup and discovery service is also context-dependent. For example by allowing mobile user to lookup services in the current environment.

- *Extensible* – Clinical applications using new context information and acquisition methods will constantly be deployed in a hospital. Therefore, a context-awareness infrastructure should be extensible in several ways. First, it should be possible to deploy, modify, and remove context services. Second, the infrastructure should support evolvement of supported types of context by dynamically load context definitions, functionality, and acquisition mechanisms, like new context sensors. Furthermore, because a hospital is in operation 24x7x365 a context-awareness infrastructure needs to run continuously. Hence, extension of the infrastructure should be done dynamically without re-starting it.

3.2 Application Programmer Interface

The goal of the framework's API is to make it easy to design and develop context-aware application for specific purposes and usage settings. This can be achieved by offering abstract modeling mechanisms, which can be implemented to fit different needs, and by having the framework contain generic implementation of such abstract modeling mechanisms. Let us consider the overall design principles for the programming API.

- *Semantic-free modeling abstractions* – The type of context information that is relevant to model and handle varies across application settings. For example, in a hospital, items like beds, pill-container, and medicine are important context information for the work of clinicians, but this is specific to hospitals. Hence the application programmer should be able to model and handle context data specific for various settings. This can be achieved by the framework by providing abstractions that make no assumption of the context data.

- *Context Transformation* – 'Context' is often derived from monitoring various aspects of the physical and digital environment, and merging these into a coherent whole. Thus, context must often be transformed into something useful for a specific application. 'Transformation' is a very general concept, which includes transformation types like interpretation or aggregation of context data [10]. Such transformers can be application-independent as well as application-dependent. A clinical example of the latter is a transformer mapping between a surgical instrument used by the surgeon to a step in a standardized operating procedure. To know where the surgeon is in the operation is highly valuable context information for a system trying to display relevant medical data during an operation. The framework should support the development of such transformers and support ways of (re-)using them separately or linked together.

- *Context Quality* – Applications are concerned with the quality of context information, including uncertainty. The clinical application trying to find relevant patient data during an operation might suggest to show more than one piece of medical data, if the uncertainty is too high. Quality measures for context information must hence be preserved from its measurement, through any transformation, and to its use by applications. Hence, the context monitors and transformers should preserve context quality appropriately.

- *Support for Activities* – The reason for capturing location and other context information is typically not for direct use in applications but to enable the reasoning at the level of *user activities* [13, 8]. For example, we want an EPR to show the correct medicine schema when the nurse is giving medicine to the patient. The framework must hence provide handles for writing application-specific code, which can 'translate' changes in the context into suggestions for user activities.

4. CONCLUSION AND FUTURE WORK

This paper has discussed the use of context-awareness in hospitals and argued for the benefits of making clinical computer applications context-aware. The paper presented examples of a context-aware EPR, a context-aware hospital bed and a context-aware pill container. Based on our evaluation of such applications we conclude that making clinical applications context-aware may be a key ingredient in creating more useful computer support for clinical work and hence enable computers in hospitals to move out of the offices and into the clinical work.

Based on this work, the paper outlined some of the overall design principles for enabling context-awareness in hospitals. A central design principle is to make context-aware services for context acquisition, handling, distribution, and consumption available as a general purpose infrastructure within a hospital. Among other things, this infrastructure should be extensible by providing support for new types of context services on runtime, and should be secure, because health-related data are sensitive. The programming interface for the developers of context-aware applications should allow a semantic-free modeling, while providing basic mechanism for transforming context information into reasoning about user's activities.

We are currently generalizing our experiences (and code) from creating the context-aware application described in this paper into the *Java Context-Awareness Framework* (JCAF) [3]. This framework addresses the design principles described above and is intended to be a general purpose framework for the design, development and deployment of context-aware computer applications. We plan to continue to use the JCAF framework in our further research and educational activities.

Acknowledgments

The Danish Center of Information Technology (CIT) and ISIS Katrinebjerg funded this research. Henrik Bærbak Christensen and Claus Bossen were much involved in the early discussions on context-awareness in hospitals.

Biography

Dr. Bardram's main research areas are pervasive and ubiquitous computing, distributed component-based system, computer supported cooperative work, human-computer interaction, and medical informatics. His main focus is currently 'Pervasive Healthcare' and he conducts research into technologies of future health – both at hospitals and in the patient's home. Currently, he is managing a large project investigating technologies for "The Future Hospital", which includes (among other things) embedding 'intelligence' in everyday artifacts within a hospital, such as in the walls of the radiology conference room, in the patient's bed, in the pill containers, and even into the pills.

Dr. Bardram received his PhD in computer science in 1998 from the University of Aarhus, Denmark. He holds a MSc in computer science and a BA in psychology. He currently directs the Centre for Pervasive Healthcare at the University of Aarhus [7].

5. REFERENCES

[1] G. D. Abowd. Software engineering issues for ubiquitous computing. In *Proceedings of the 21st international conference on Software engineering*, pages 75–84. IEEE Computer Society Press, 1999.

[2] J. E. Bardram. Designing for the dynamics of cooperative work activities. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 89–98. ACM Press, 1998.

[3] J. E. Bardram. Design, Implementation, and Evaluation of the Java Context Awareness Framework (JCAF). Technical Report CfPC 2003–PB–??, Center for Pervasive Computing, Aarhus, Denmark, 2003. Available from http://www.pervasive.dk/publications.

[4] J. E. Bardram and C. Bossen. Moving to get aHead: Local Mobility and Collaborative Work. In K. Kuutti, E. H. Karsten, G. Fitzpatrick, P. Dourish, and K. Schmidt, editors, *Proceedings of the Eighth European Conference on Computer Supported Cooperative Work*, pages 355–374, Helsinki, Finland, Sept. 2003. Kluwer Academic Publishers.

[5] J. E. Bardram, C. Bossen, A. Lykke-Olesen, K. H. Madsen, and R. Nielsen. Virtual video prototyping of pervasive healthcare systems. In *Conference proceedings on Designing Interactive Systems: processes, practices, methods, and techniques (DIS2002)*, pages 167–177. ACM Press, 2002.

[6] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison-Wesley, 1998.

[7] Center for Pervasive Healthcare, University of Aarhus, Denmark. http://www.cfph.dk.

[8] H. B. Christensen. Using logic programming to detect activities in pervasive healthcare. In *International Conference on Logic Programming ICLP 2002*, Copenhagen, Denmark, Aug. 2002. Springer Verlag.

[9] H. B. Christensen and J. E. Bardram. Supporting human activities – exploring activity-centered computing. In G. Borriello and L. E. Holmquist, editors, *Proceedings of Ubicomp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, pages 107–116, Göteborg, Sweden, Sept. 2002. Springer Verlag.

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

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

[12] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In M. Naghshineh and F. Mattern, editors, *Proceedings of Pervasive 2002: Pervasive Computing : First International Conference*, volume 2414 of *Lecture Notes in Computer Science*, pages 167–180, Zürich, Switzerland, Aug. 2002. Springer Verlag.

[13] J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*. IEEE Computer Society Press, 2002.

[14] F. Hohl, L. Mehrmann, and A. Hamdan. A context system for a mobile service platform. In H. Schmeck, T. Ungerer, and L. Wolf, editors, *Proceedings of ARCS 2002: Trends in Network and Pervasive Computing*, volume 2299 of *Lecture Notes in Computer Science*, pages 21–33, Karslruhe, Germany, Mar. 2002. Springer Verlag.

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

[16] T. Moran and P. Dourish. Introduction to this speical issue on context-aware computing. *Human-Computer Interaction*, 16:87–95, 2001.

[17] R. Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser. An overview of the parctab ubiquitous computing environment. *IEEE Personal Communications*, 2(6):28–43, 1995.

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