Context Information vs. Sensor Information: A Model for Categorizing Context in Context-Aware Mobile Computing

Louise Barkhuus
Department of Design and Use of Information Technology
The IT University of Copenhagen
Glentevej 67, DK-2400 Copenhagen, Denmark
barkhuus@it.edu

Keywords: Context-aware computing, context modelling, ubiquitous computing, mobile computing, mobile collaborative work.

Abstract

Context-aware computing proposes technology that adapts according to the user's situation and environment. Within mobile collaborative work the situation is viewed as an important factor, here context-awareness attempts to facilitate coordination in several ways. Although much research has been conducted, context-awareness is still defined very differently. Besides differing in definitions of context, most research fails to separate between complex context information and precise sensor information. Knowing that collaborative work can benefit from some degrees of context-awareness and that context-awareness is essential to a ubiquitous computing environment, this paper focuses on initial design of context-aware applications. After reviewing past categorization models of context, The Context Information Model is proposed. Categorizing from the user's view point, the model separates strictly between context information and sensor information. It is then exemplified how the model can be applied to an actual consumer device. The paper concludes that if context-aware computing applications are to be developed for real collaborative and consumer settings, sensor measures should be gradually incorporated to comprise a simplistic categorization of context rather than define context in detail.

1. INTRODUCTION

In collaborative work, mobility of people as well as artifacts is often relied upon as a facility to improve coordination. The mobility constitutes the flexibility needed for collaborative work and it is critical to collaborative work [6]. It is important to communicate and interact in order to collaborate and mobile computing offers a great range of possibilities for this. To make mobile devices more suited for the demands of the participants of collaboration, the issue of context-awareness is relevant. It describes mobile applications that change according to work situation and collaboration context and thereby provide the users with a more flexible work sphere.

Context-aware computing has been an active research area for over a decade without leaving any real-life applications on the consumer market. From the PARCTab experiment of the early nineties [7] to the more sophisticated tour-guides utilizing GPS [1, 4], the field has yet to climb outside the research laboratories and provide consumers with active autonomous applications for their mobile devices. The research of context-awareness is not lacking in amount and several different approaches have been taken when dealing with the subject. From creating a widget model [5] and describing a blackboard model [13], to characterizing context into several classes [3, 11], many researchers have attempted to structure the complex nature of context sensing into categorizations and models. Still, none of these models are able to grasp the actual human understanding of the diversity of context information as distinguished from the technical understanding of sensor information. Most of the models take a technical approach, arguing for ways to develop context-aware computing that relies on sensor information as the primary source of context, regardless that context categories often consist of several pieces of sensor information. The field is in need of a common understanding of the relation between context as a broad concept and actual sensor information. The development of context-aware applications will benefit from a more strict distinction between the two kinds of information, which will support the initial design of application as well as the analysis of existing context-aware technology. By drawing on past research and present context categorizations, this paper suggests a model, The Context Information Model, separating the context information from the sensor information, and thereby facilitating the design process of context-aware applications. As a future perspective and to support further understanding of context within context-aware mobile computing, the model should eventually provide insight into the analysis of existing context-aware technology.

2. CONTEXT-AWARE COMPUTING

The concept of context-aware computing describes infrastructures that include sensor information from its surrounding environment, physical as well as electronic, in the computing applications. The collected information creates part of the *context* that comprises the interaction between humans and computing units; the applications change according to their present context. In collaborative work, context is an important issue, as coordination seems to depend significantly upon the participants' context. An example of a piece of relevant context information is the most common question posed when communicating on mobile phones: “where are you?”. Also the increased specialization of computing devices make independent context variables essential to mobile collaboration technology, which also supports the vision of ubiquitous computing. Context-awareness is the attempt to relate the technology to the actual collaboration situation and environment it is part of.

The definition of context differs depending on theoretical approach. The concept is often described according to sensor information or as a broader concept regarding the overall environment. Although the concept is used widely in the area of ubiquitous computing, it seems that no standard definition of it exists. When reviewing the literature within the area, the abstraction of the concept is evident; from using context-awareness to describe a simple location-aware system to requiring utilization of several environmental measures, the concept is in dire need for at least a common understanding. In order to grasp the complexity of context in relation to computing technology, the research within context-awareness is reviewed and summed up in a definition that should work as the foundation for the proposed model.

2.1 Defining Context-Awareness

The first to actually use the word context-awareness were Schilit and Theimer. In 1993 they introduced the term *dynamic customization*, which describes a new kind of application capable of adapting during a session and, although not being the first context-aware application, the notion of such attribute probably inspired the term *context-aware computing* introduced a year later in 1994 [10, 9]. Schilit and Theimer defined it as “the ability of a mobile user’s applications to discover and react to changes in the environment they are situated in” [9]. While this is a very broad definition they narrow it by defining context information as location, identities of nearby people and objects and changes to those objects. However, the actual context information that Schilit and Theimer use in their Active Map Service is limited to location, perhaps because of technical limitations. They suggest that future context-aware applications should include much more than location information, a suggestion that numerous researchers have attempted to build upon.

Brown et al. define context-awareness simply as applications that change behavior according to the user’s context. They describe how laboratory context-aware applications can be used in real settings, by developing *stick-e notes*, a generalized way of representing context written in a standard notation (SGML), as part of their context paradigm [2]. Acknowledging that most context-aware applications only utilize location as context, the stick-e note should facilitate a more broad use of context. Their description of context information is thus not limited but examples that are given include temperature, season of the year as well as the information defined by Schilit and Theimer. The context information is relevant for the understanding of context-awareness, therefore the description of context measures is a common way of defining it. Some researchers solely provide examples of context as a definition of context-awareness (eg. [5, 11]). The lack of a comprehensive definition is also a source of conflicting definitions, meaning that one application is considered context-aware by some researchers but not by others.

Striving towards a useful definition, one of the most simple and ample definitions of context-awareness is still the one provided by Schilit and Theimer. Limited to the *discovery and reaction to changes in environment*, it captures the essence of a reactional system that takes environmental measures into account [9]. Because it includes ‘discovery’, it also limits the applications in question to the ones that change independently of direct user input contrasting the applications that change according to predefined modes and personalization such as profiles, which in some cases are also referred to as context-aware (eg. continuous context-aware application as defined by Brown et al. [2]). Schilit and Theimer’s definition of context-awareness is closely related to the definition proposed in this paper. The proposed definition here, claims that context-awareness is *an applications ability to detect and react to environment variables autonomously*. However, as environment and situation are the most commonly used synonyms for context within the research area [5], the concepts only define the overall notion of context-awareness, context is not defined in a more thorough matter. Therefore context is often defined by categorization of context and sensor information, which is the essence of context-awareness and also this paper’s further approach.

3. CATEGORIZATIONS OF CONTEXT

The categorization of context information is common for a large part of research within the area and aims to supply the definition of context-awareness with a more clear understanding. However context is not a static construct and therefore leaves many researchers with a categorization scheme that is not necessarily useful for the development of real applications. It seems reasonable to categorize context theoretically to get further insight into the concept, but it means that the classification schemes are foremost used to clarify the diversity of context information in general. One example is the distinguishing between computing factors (networks connection, user interface size etc.) and human factors (identity of user, social situation etc.). Some categorizations within context-aware computing attempt to classify the actual applications [2, 8] as for example *active* or *passive* applications [3]. The active application is one that automatically adapts to context measurements and thereby changes the application’s behavior, where the passive application only presents new or updated information to the user. These categorizations strive to define criteria for determining if an application is actually context-aware and how different levels of context-awareness can be applied to applications.

3.1 Environmental Categorization

The most common way of categorizing context is to classify the information into separate environmental clusters. Basing their categorization on Schilit et al. for example [8], Dey et al. divide the environment into three levels: Computing context (eg. network connectivity and nearby resources), user context (eg. user profile, location and people nearby) and physical context (eg. lighting and noise level) [5]. Schmidt and Forbess, on the other hand, only divide context environment into two categories, human factors and physical environment, where the physical environment consists of Dey et al.’s computing context as well as their physical context [12]. These categories have a set of features and these features each have a range of values. The model is created to be exhaustive in nature; the sensor information and its values are the last step in the categories. This way the model can be adapted according to a specific application and its purpose.

Most of the categorizations do not consider non-environmental factors that often make up the larger context, such as the user's internal state or the task that is to be carried out. The fact that a situation is fluid and ever-changing in nature and that it cannot always be predicted from small pieces of sensor information is only sparingly mentioned in this relation. It is an important realization when attempting to develop context-aware applications, but even though the non-environmental measures are not easily categorized, the modelling of physical factors can be useful for a greater understanding of the limited possibilities of using context information.

3.2 The Utilization of Categorization in Actual Applications

Although Schmidt et al. classify context in two categories (human factors and physical environment) the categorization is not applied to their sensor fusion that is proposed as a tool for acquiring context information [11, 12]. Instead the sensor fusion proposes a four layered architecture, which provides the levels of abstraction necessary to implement context-awareness (sensors, cues, contexts and scripting). The approach is detailed but actual end-use of the applications is not considered in contrast to the Context Information Model. Apart from Schmidt et al., the reviewed categorizations of context do not separate context information from sensor information. Chen and Kotz mention the difficulty of acquiring *high-level contexts* (eg. social situation) and relate it to low-level sensors by suggesting to combine several pieces of information to create a high-level context [3]. The concept is interesting but not illustrated by examples of actual implemented applications.

One categorization that is utilized to develop an actual application is one used in the GUIDE project presented by Cheverst et al. [4]. Although mainly using location information, the authors categorize information into object types: *navigation point objects* and *location objects*. Location objects are the actual located physical objects such as a specific building and navigation point object are the way-points between location objects that assist in navigation.

The categorizing of context information aims to provide a more clear understanding of the broad concept and of which contextual measures could be utilized in context-aware computing. Most categorizations are derived from earlier ones (eg. [5] and [3]) and classify the context not in relation to sensor information but in relation to a notion of what the application should know. Dey et al.'s categorization of context, for example, is not very clear [5]: The *network capacity* information that is part of the computing environment is sensor specific, meaning it is measurable, where the *social situation* that is part of the user environment is situation specific, meaning that it needs a human definition (perhaps comprised of numerous pieces of sensor information) before it can be measured.

3.3 Limiting the Sensor Information

Although the categorization of context information enables a more structured thinking of context-awareness, only in rare cases will more than a handful of separate context measures be used in actual applications. When designing an application it is therefore only relevant to define context for the specific purpose of the application. Research shows that even though much effort has been done to use as much information as possible, the cumbersomeness of the implementations increases dramatically with every new piece of sensor information included. Relating to the unpredictability of situations, it is likely that more context measures do not necessarily provide an application with a more precise definition of a specific context. Since overall context is not very predictable in nature, the information that comprises the situation should be limited and the possible actions that the application take limited as well. Many of the reviewed categorizations of context lack the consideration that sensor information should be limited in relation to the end-use of the application. Categorizations of context would therefore benefit from a more application specific perspective, one that the model of Context Information attempts to provide.

4. THE CONTEXT INFORMATION MODEL

The Context Information Model is developed to support the initial design of context-aware applications as opposed to other categorizations that facilitate a better understanding of what context is. Eventually it can be expanded to provide a framework for analyzing context-awareness in present technology but the primary purpose is the implementation of context measures in mobile devices. When developing mobile computing to support collaboration, the context information is essential to the success and the quality of the work. However, in order not to confuse or annoy the user, the context information and the sensor information should be limited and clearly separated so that the actual situation is not compromised by overly detailed or irrelevant context categories. Where other researchers categorize context in broad classes, such as human, physical or computational factors, the Context Information Model categorizes the actual *use* of the device or application before drawing out the context information needed in the specific use. This approach enables the vague measures of environment and situation to be modelled according to the specific task that the application is developed for.

The model emphasizes a strict separation between context information and specific sensor information; the separation enables the development of context measures that are limited and clearly defined. The distinction is often overlooked by research and the two kinds of information are treated similarly in categorization models. When discussing location for example, it is necessary to differ between relative location (defined as 'in that specific office' or 'in that user's car') and actual position (the sensor information retrieved by for example the GPS system, such as x degrees north by y degrees west). The Context Information Model introduces three levels of contextual information: *user level*, *context information level* and *sensor information level*.

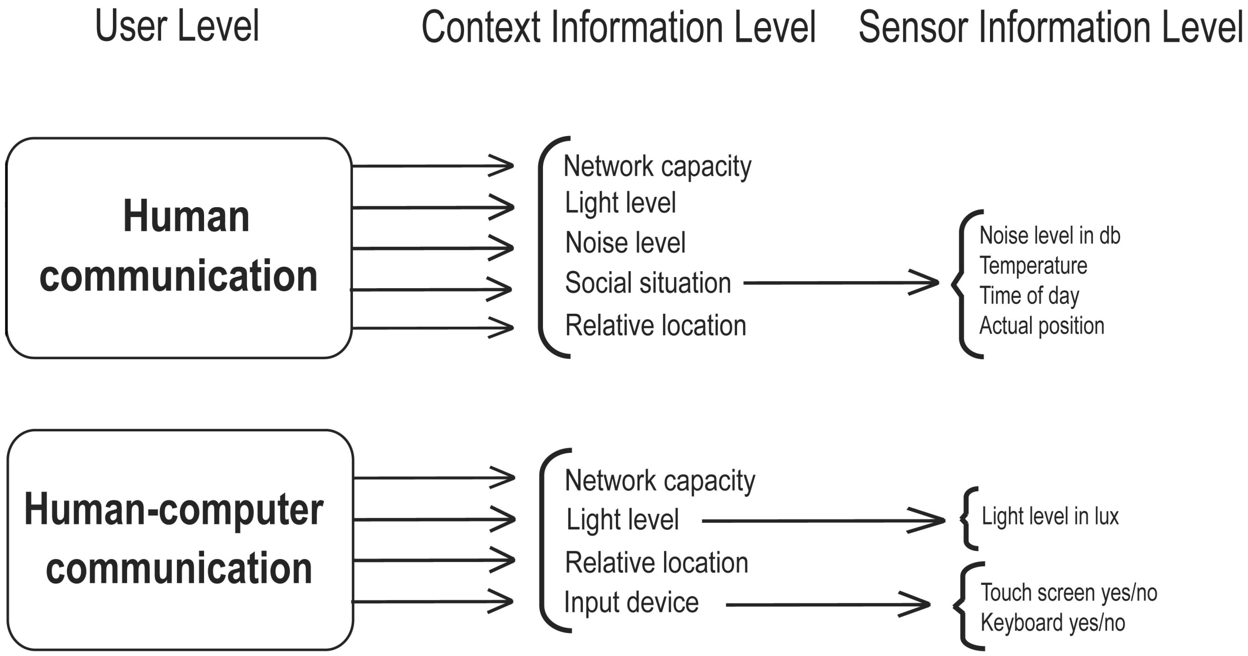

Figure 1: The Context Information Model.

User Level
The user level is the actual task that the device or application is developed for such as *human communication* and *human-computer communication*. Human communication includes voice communication, data communication (eg. text-messages and e-mail) and visual communication (eg. video conferences). Human-computer communication includes search for information, over a network (eg. the Internet) and within the device (eg. search for a phone number in a cell-phone's address book), as well as information input to either network services or internal use within the application (eg. a personal agenda). Note that the general model is not created to be exhaustive in nature and more user levels could be added in order to expand it. As will be shown in the application of the model, most tasks fall within these two groups, but it is likely that more user levels exist. However, the overall purpose of the model is to describe context in a more specific manner, therefore only a limited scope of tasks are included in the general model.

Context Information Level
The context information level consists of the context information that is relevant for the user level. The context information is defined in a complex manner as it is general context, which is to be described before the applicable sensor information can be found. The context information is limited to the information relevant for the use since excess context information would make the design of applications increasingly complex without the real benefit of context-awareness. An irrelevant piece of context information, for example, for a human communication situation is the user's identity. Because mobile devices are personal (at least the vast majority), the context is centered around the specific user and the information is something that does not change [1]. Each level is a number of predefined classifications for each user level where most context information will be described in words rather than exact measures. The predefined classifications of social situation could for example be 'meeting', 'outside', 'general day' or 'night'. All actual situations should then fall within these predefined but fairly general categories, a factor often proposed within context-aware computing; most early research dealt with a very limited number of distinct situations.

Sensor Information Level
The sensor information level consists of the specific sensor information necessary to determine the elements of the context level. Depending on both user level and context information level, the sensor information is what the device or application needs in order to become context-aware. At this level it is not relevant, for example, to describe information that can be measured in an ambiguous way; location should be described as exact position where the relative location would belong to the context information level. A piece of context information can consist of several pieces of sensor information. The complex example of social situation is comprised by noise level, time of day compared to predefined agenda, temperature and actual position. These pieces of sensor information are in the example of human communication enough to distinguish (however not extensively) between the predefined social situations.

[1]This lack of user identity information in the general model should not be confused with the application's knowledge of *other* users' or devices' identity.

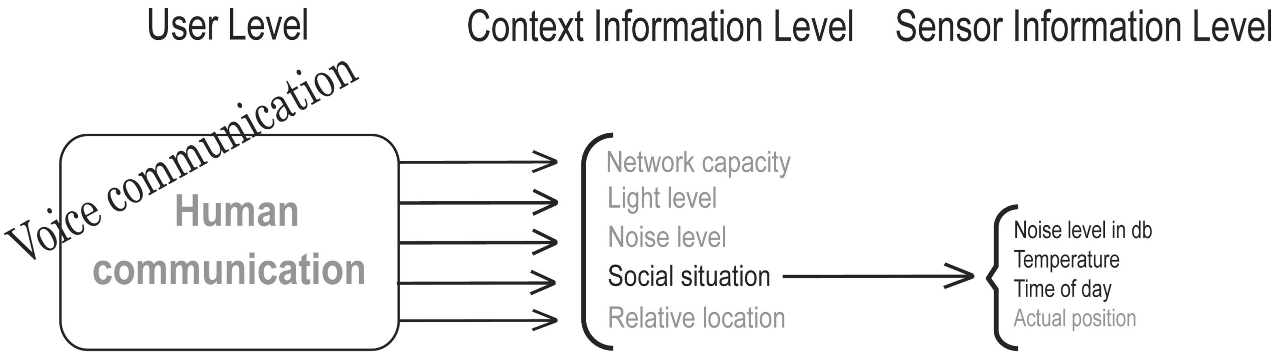

Figure 2: An example of The Context Information Model applied to a static function.

4.1 The Applied Context Information Model

The model attempts to illustrate how the complexity of context can be limited in scope and implementation by considering only relevant context-information. Actual context as a setting for collaboration differs highly according to non-measurable factors such as the user's internal state. The model does therefore not attempt to solve greater problems of what actually constitutes more complex contexts, such as social environment or surrounding people and objects' identity. It merely suggests a method of defining sensor information measures that could add to an understanding of context in real life context-aware applications. The Context Information Model is developed to design static functions as well as dynamic functions, which the two examples should illustrate. The static functions are those that run continuously on the device, where dynamic ones are specifically applied to specific tasks that need to be fulfilled.

Example of an Applied Static Function

To exemplify the utilization of The Context Information Model for actual consumer applications, a cell phone with context-aware features is proposed. The cell phone is developed with the capability to measure sensor information according to the context and thereby act according to the social situation. Since the primary use of a cell phone is human communication, context-aware features could include the device's ability to, on the basis of social situation, adjust the volume of the speaker as well as ringing tone. Figure 2 illustrates the applied model for this application. The predefined categories of the context level in this example are limited to three relying on noise level, temperature and time of day as sensor information. The social situation context categories would in this example consist of 'high', 'regular' and 'discreet'. A high noise level during the day will result in the 'high' context category where a medium noise level results in a 'regular' category. However if the measured temperature is lower than 15 degrees Celsius (60 degrees Fahrenheit), the context category would always be set to 'high' because the device very likely would be outside. The third context category is the 'discreet' category that is also applied during the night. As this is a simplified example, the context categories are not utilized to the furthest extent but only attempts to indicate a possible situation based on few pieces of sensor information. Note that the user level of the applied as well as the general model does not imply the actual action that the device/application should perform; this factor is not part of the model. Instead the task is a general task that the user needs to perform. The context information level solely consists of context information that would be relevant for the specific task. Therefore, the criteria in the context information level can be limited in reality as all information might not be necessary to make the application perform the desired action.

Example of an Applied Dynamic Function

The example of a dynamic function could be applied to any mobile device. Dynamic functions could be specifically related to the device's use and application, as in this example of an applications ability to 'find the nearest of predefined friends', seen in figure 3. The function of 'finding the nearest friend' is part of *Human-computer communication* and would rely on the context information of *relative location*. The context information of relative location is determined by sensor information measuring the user's actual position, the predefined friends' identity and the actual position of these friends. In the example of a dynamic function such as this, the *criteria* of the function consists of the context and sensor information, contrasting the static functions where the context level is divided into several context categories.

Limitations of the Model's Applicability

These examples of the model's application are very simplified and the the model still only offers limited opportunity to relate specific functions of the application to the user tasks. The tasks are termed in a general manner but when applying the model to the design of a real consumer application, the function should be specified within the boundary of the user level. Functions that lie within the scope of control systems (systems that control features within the device without the user involvement), such as mobile phones' automatic change between cell base stations, are not part of the proposed model. The important distinction that the model emphasizes however, is the separation of the general notion of context information from the actual sensor information. It is not claiming that general context is found from few pieces of sensor information, it is merely suggesting a method

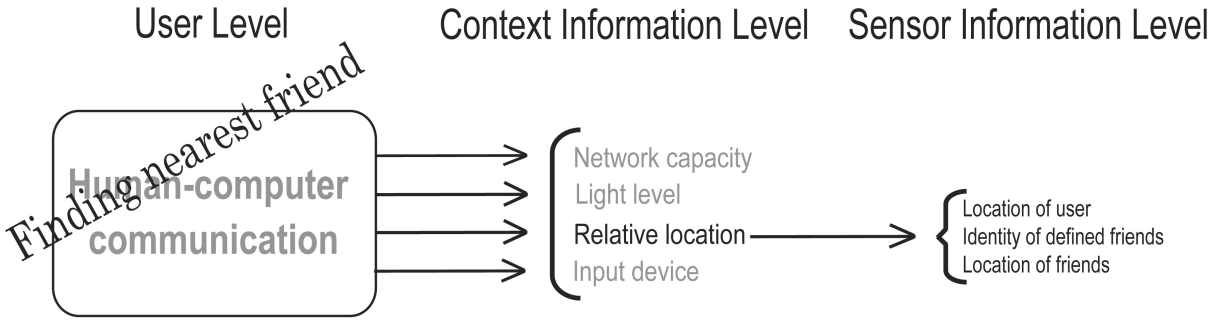

Figure 3: An example of The Context Information Model applied to a dynamic function.

where sensor information can be utilized to support dynamic as well as static functions of mobile applications. Eventually, the model could be applied to more complex functions of device applications, which should be subject for future research.

4.2 The Model's Relation to Other Models

Most of the previously reviewed categorizations and models for developing context-aware applications take a technical approach such as Dey et al.'s three categories of context (computing, user and physical context) [5] or present technical specifications such as Cheverst et al.'s GUIDE architecture [4]. The Context Information Model takes context-awareness one step back and views context from the user's perspective in order to offer a suggestion for a limited variety of context information. Many of the context-aware applications developed in research labs also use a limited set of context and sensor information but their reasons for this often seem to be due to technical constraints rather than a conscious limitation made regarding the end-use of the product. A model such as The Context Information Model is therefore not directly comparable to the previous reviewed categorizations. The model is not attempting to cover the more technically oriented issues of context and sensor measuring or the more wide definition of context. It is merely one suggestion to how context can be applied in actual consumer applications. In contrast to other categorizations the model is application- and task specific, leaving it vulnerable to future applications that could possibly make certain parts irrelevant. The non-exhaustive and flexible nature however, enables it to adapt and hopefully support new applications.

5. CONCLUSIONS

Collaborative work often requires a sense of situation, which context-awareness strives to supply. While context is not a static measure, the sensor information that is available can to a certain extent aid mobile applications used for coordination. This paper has reviewed the modelling and categorization of context within context-aware computing in order to find if a common understanding of context-awareness exists. Finding the categorizations of context to be lacking in differentiation between context information and sensor information, The Context Information Model is proposed as a means of designing context-aware applications from a limited amount of context measures. Lastly, examples of the model's application to consumer devices are presented, leaving more complex user functions for later expansions of the model.

Although context is a highly complex matter that is difficult to trace or determine, by considering individual measures it should be possible to create usable interactive applications, which aid collaborative work. While the Context Information Model is not an attempt to trace the general context or situation of the users, it should support real-life applications that take a limited number of sensor measures into account. The presented model also offers potential for further development in that comprehensiveness is not strived for in its present state. The analysis of present context-aware applications is for example one area that should be considered in future revision and development of the model. Further expansion also includes the actual design of a context-aware application on basis of the model, which would enable insight to its usefulness beyond initial context design and definition. The model consists of features that should allow it to scale up or down to other potential applications; the potential exploration of context-awareness should prove the employment of such applications to be of great benefit to the future users of mobile devices.

Context-awareness is one of the more complex areas of mobile computing and the differences in definition of context is source of great differences of approach. Although much research is based on previous researchers' results, the variety of approaches is impairing future research within context-aware mobile computing. Context-awareness is defined in different terms depending on the overall area focused on such as for example ubiquitous computing as opposed to collaborative studies. The challenge lies within a common understanding of not *how* to define context but in the treatment and definition of sensor information as *part* of context information. However, the diversity of research also results from the early state that context-aware computing occupies; evidently not many applications, research based as well as consumer oriented, take advantage of more than a limited number of context measures (often only one: location). The increasing complexity of sensor technology and the possibility of tracing relevant context measures, make context-awareness a challenging and large scale research area, especially in relation to mobile collaborative work. If the goal is to create a transparent and ubiquitous computing environment, context-awareness is a key factor for new consumer targeted applications. The area possesses great potential but in order to provide the consumer market with actual applications, future research should focus on gradually incorporating context measures into applications as well as finding a common categorization of context and sensor information.

6. REFERENCES

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

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

[3] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Paper TR2000-381, Department of Computer Science, Darthmouth College, November 2000.

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

[5] A.K. 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(2–4):97–166, 2001.

[6] P. Luff and C. Heath. Mobility in collaboration. In *Proceedings of the ACM 1998 conference on Computer supported cooperative work*, pages 305–314. ACM Press, 1998.

[7] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The parctab mobile computing system. In *Proceedings of the Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 34–39, Napa, California, October, 1993. IEEE Computer Society.

[8] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications*, Los Alamitos, CA, 1994. IEEE.

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

[10] B. Schilit, M. M. Theimer, and B. B. Welch. Customizing mobile applications. In *Proceedings of the USENIX Symposium on Mobile and Location-independent Computing*, pages 129–138, Cambridge, MA, 1993.

[11] A. Schmidt, M. Beigl, and H.W. Gellersen. There is more to context than location. *Computer and Graphics*, 23:893–901, 1999.

[12] A. Schmidt and J. Forbess. What GPS doesn't tell you: Determining one's context with low-level sensors. *The 6th IEEE International Conference on Electronics, Circuits and Systems, Paphos, Cyprus*, September 5–8 1999.

[13] T. Winograd. Architectures for context. *Human-Computer Interaction*, 16(2–4):401–419, 2001.

Biography

Louise Barkhuus is a Ph.D. student at The IT University of Copenhagen, where she also obtained her M. Sc. in Computer Science in 2001. She is currently researching context-awareness, and in particular the design of context-aware applications for mobile devices.