ISSN: 2321-7782 (Online)
Volume 3, Issue 7, July 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Context Awareness in Mobile Computing

Komal Tayde[1]
Department of Computer Science
Anuradha College of Engineering,
Chikhali, India

Prof. Amit Bhala[2]
Department of Computer Science
Anuradha College of Engineering,
Chikhali, India

Abstract: Context awareness is a growing area in the field of ubiquitous computing. It is particularly related to the emerging sub-field of mobile computing as a user's context is changing more rapidly when a user is mobile, and interacting with more devices and people in a greater number of locations. Context awareness nowadays is gaining applicability in interactive mobile computing systems. To implement this concept of context awareness in design of applications for dynamic environments, it is necessary to understand context and context-aware development frameworks and what constitutes context awareness.

In this report, we describe better understanding of context awareness, design principles of context aware applications, their usability. Also we describe some of the risks in building context aware applications and the solutions to address these. Along with this, we provide few guidelines for developing mobile context aware applications and discuss future directions on context aware applications.

Keywords: context; context awareness; ubiquitous computing; mobile computing.

I. INTRODUCTION

In past few decades, use of mobile devices such as mobile phones and personal digital assistants has spread rapidly in the world. Fast growing technology has made them affordable and available to large group of users. Today users of computing devices are faced with diverse devices either mobile or fixed featuring diverse interfaces and used in diverse environments. This is the ‘third wave of computing'- a step towards realization of a ubiquitous computing paradigm, where specialized devices outnumber users. Mobile computing is a fully realized phenomenon of everyday life and is the first computing platform that is truly ubiquitous. Technical enhancements in mobile computing, such as component miniaturization, enhanced computing power, and improvements in supporting infrastructure have enabled the creation of more versatile, powerful, and sophisticated mobile devices.

The concept of context-aware computing was first introduced by Mark Weiser in a seminal paper ‘The Computer for the 21st Century' (Weiser, 1991). He describes ubiquitous computing as a phenomenon ‘that takes into account the natural human environment and allows the computers themselves to vanish into the background.' He also shapes the fundamental concepts of context-aware computing, with computers that are able to capture and retrieve context-based information and offer seamless interaction to support the user's current tasks, and with each computer being able to ‘adapt its behavior in significant ways' to the captured context. Schilit and Theimer (1994a) first introduce the term context-aware computing in 1994 and define it as software that ‘adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time'.

We prefer a more general definition of context and context-awareness: Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves, and by extension, the environment the user and applications are embedded in. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

Different perspectives on how mobile applications can take advantage of context have been advanced. Thus applications can automatically adapt their behaviour according to discovered context (active context), or present the context to the user on the fly and/or store it for the user to retrieve later (passive context). This has led to context-aware computing, defined in two ways: firstly, active context awareness automatically adapts to discovered context by changing the application's behaviour; and secondly, passive context awareness presents the new or updated context to an interested user or makes the context persist for the user to retrieve later. Active context-aware computing leads to new applications, especially in mobile computing, and requires more infrastructure support to help eliminate unnecessary user cooperation and make the technology as ‘calm' as possible.

II. Design Principle

Dey defines context as any information characterizing a situation related to the interaction between users, applications and the surrounding environment. He proposed use of conceptual models and tools to support rapid development of context-aware applications that could better inform empirical investigation of interaction design and social implications of context-aware computing. This work attempted to enable a new phase of context-aware application development with the intention of helping applications developers understand what context is and what it can be used for, and to provide concepts and practical support for software design and construction of context-aware applications. We represent few design principles that should be considered while developing context aware application.

Context-aware systems can be implemented in many ways. The typical approach considers a number of special requirements and conditions, such as location of sensors (local or remote), number of possible users, available resources (such as high-end personal computers or small mobile devices), and extensibility of the system. Context aware applications are generally based on Context Aware models. Strang and Linnhoff-Popien summarized the most relevant context-modeling approaches based on data structures used for representing and exchanging contextual information in their respective systems. Few of them are listed below:

Key-value models: These represent the simplest data structure for context modeling. They are frequently used in various service frameworks, where key-value pairs are used to describe the capabilities of a service. Service discovery is then applied with matching algorithms which use these key-value pairs.

User-context perception model: This is a model created to help the designer understand the challenge(s) faced in creating context-aware systems. As an example, a car navigation system works very well if one is in a new city; however, when using it around a familiar area one may sometimes be surprised at the route it tries to direct one to.

Mark-up models: These use a hierarchical data structure comprising mark-up tags, attributes and content to create profiles which represent a typical mark-up scheme model.

Graphical models: A number of approaches have been proposed where contextual aspects are modeled using Unified Modeling Language.

Object-oriented models: Modeling context using object-oriented techniques offers the full power of object orientation (e.g. encapsulation, reusability and inheritance). Existing approaches use various objects to represent different context information (such as temperature, location, etc.), and encapsulate details of context processing and representation. Access to the context and context-processing logic is provided by well-defined interfaces like the hydrogen model.

Logic-based models: These models have a high degree of formality, and typically facts, expressions and rules are used to define a context model. A logic-based system is used to manage the aforementioned terms and allows addition, updating or removal of new facts. The inference (also called reasoning) process is used to derive new facts based on existing rules in the systems. Contextual information is then represented in a formal way as facts.

Ontology-based models: Ontology represents a description of concepts and their relationships. These models are very promising for modeling contextual information due to their high and formal expressiveness and possibilities for applying ontology reasoning techniques.

Example of Context Aware Model

To explain this phenomenon with this model, we assume the context-aware system (right side in Fig. 1) is of equal quality in both locations; this means that the difference must be on the user's side. The sensory perception (e.g. visual matching of buildings and places you know, based on your sight) is different in familiar and new places. In the new place you lack reference points, and the memory and experience parts in the model differ significantly. In the familiar environment you will have expectations about which route to take and which would be a good choice. In the unfamiliar environment you lack experience and reference points, and hence your expectation is simply that the system will guide you to your destination. The result is that a navigation system that successfully guides you to your destination with a non-optimal route will satisfy your expectations in an unfamiliar environment, but be frowned upon in a familiar environment. In the familiar environment we have a substantial awareness mismatch, whereas when navigating in new surroundings we have minimal awareness mismatch.

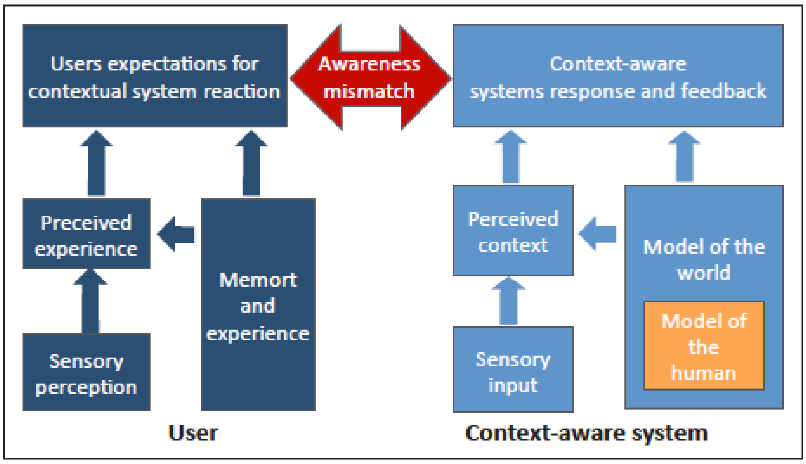

Fig. 1 The user context perception model

The quality of context-aware systems, as perceived by the user, is directly related to the awareness mismatch, and a good design aims at designing systems with minimal awareness mismatch. A prerequisite for creating a minimal awareness mismatch is that the user understands what factors have an influence on the system. In the example of a simplistic car navigation system, this factor is only current location and nothing else. In such a case the user knows the system's reactions are based purely on current location and destination, and the user may attribute the system's response to these factors.

III. Context Aware Applications

The first context-aware applications were centered on mobility. The Active Badge location system used infrared-based badges and sensors to determine the location of workers in an indoor location (Want et al., 1992). A receptionist could use this information to route a phone call to the location of the person being called, rather than forwarding the phone call to an empty office. Similarly, individuals could locate others to arrange impromptu meetings. Schilit, Adams and Want,(1994b) also use an infrared-based cellular network to location people and devices, the PARCTAB, and describe 4 different types of applications built with it (Schilit et al., 1994b).

This includes:

• **Proximate selection:** Nearby objects like printers are emphasized to be easier to select than other similar objects that are further away from the user;

• **Contextual information and commands:** Information presented to a user or commands parameterized and executed for a user depend on the user's context;

• **Automatic contextual reconfiguration:** Software is automatically reconfigured to support a user's context; and

• **Context-triggered actions:** If-then rules are used to specify what actions to take based on a user's context

Examples:

1. **Wristwatch:** Schmidt provides an example of designing a user interface for a wristwatch. The watch is used both indoors and outdoors, in the dark as well as in sunlight, when running to catch a bus or when attending a boring lecture. A good user interface designer will create varied user interfaces for each situation. The context-aware computing approach enables one to create a context-aware watch, where all situation-optimised designs are combined in a single design. The watch is designed so that it can recognise each of the situations, and then reconfigure itself based on the recognised context.

Fig. 2 Design sketches that illustrate time visualisations in different contexts. (a) For users running to catch a bus, making it easy to see the minutes in large fonts. (b) For boring lectures and meetings, showing a countdown to the end, with some information to engage the user. (c) Visualisation giving only a very coarse idea of the time, similar to information you get from the sun, to use for example when hanging out with friends - when time does not matter.

2. **Smart Homes:** When you're not home, nagging little doubts can start to crowd your mind. Did I turn the coffee maker off? Did I set the security alarm? Are the kids doing their homework or watching television? With a smart home, you could quiet all of these worries with a quick glance at your smartphone or tablet. You could connect the devices and appliances in your home so they can communicate with each other and with you. Any device in your home that uses electricity can be put on your home network and at your command. Whether you give that command by voice, remote control, tablet or smartphone, the home reacts. Most applications relate to lighting, home security, home theater and entertainment, and thermostat regulation.

3. **Google Glasses and wearables:** Minimum distraction is key success factor for personal devices and wearables. This is done by the next step in this digital revolution; continuous context aware communication. Knowing the context of the person using the device enables the wearable to only present the information that is absolutely relevant to the user. Some of the benefits of Google glasses (Fig. 3) are-

   - Interaction without distraction
   - Always context aware

- Eliminate multitasking
- Up-close or even embedded
- Re-thing the design for context awareness
- Just in time, just in context devices

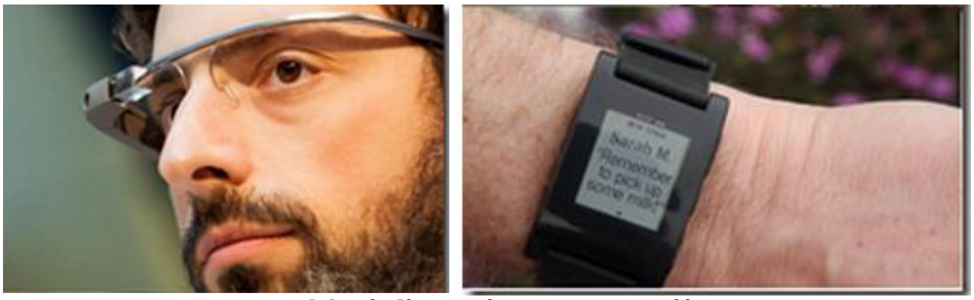

Fig. 3 Google Glasses and context-aware wearable

4. **Satellite Navigation System:** In a Satellite Navigation System (SatNav), the current location is the primary contextual parameter that is used to automatically adjust the visualization (e.g. map, arrows, directions…) to the user's current location. An example is shown in below figure. However, looking at current commercial systems, much more context information is used and much of visualization has been changed. In addition to the current GPS position, contextual parameters may include the time of day, light conditions, the traffic situation on the calculated route or the user's preferred places. Beyond the visualization and whether or not to switch on the backlight, the calculated route can be influenced by context, e.g. to avoid potentially busy streets at that time of day.

Fig. 4 Context-aware Satellite Navigation System in a car

5. **Conference Assistant**: The Conference Assistant examines the conference schedule, topics of presentations, user's location and research interests to suggest which presentations to attend. Whenever the user enters a presentation room, Conference Assistant automatically displays the name of the presenter, title of the presentation and other related information. Available audio and video equipment automatically record the slides of the current presentation, comments and questions for later retrieval.

6. **Office Assistant**: This is an agent-based application that interacts with visitors at the office door and manages the office owner's schedule. The assistant is activated when a visitor approaches, detected by pressure-sensitive mats placed on both sides of the office door. It adapts its behavior to contextual information such as identity of the visitor, office owner's schedule status and the owner's willingness to see the visitor.(Example Fig. 5)

Fig. 5 Office Assistant Agent

IV. Usability of Mobile Context Aware Applications

With context information being provided as implicit input to applications and with those applications using this context to infer human intent, there are greater usability concerns than with standard applications that are not context-aware.

To ground our understanding of these abstract concerns, Dey & Jonna we studied the usability and usefulness of a variety of context-aware applications (Barkhuus & Dey, 2003a; 2003b). We described a number of real and hypothetical context-aware applications and asked subjects to provide daily reports on how they would have used each application each day, whether they thought the applications would be useful, and what reservations they had about using each application. All users were given the same set of applications, but users were split into three groups with each group being given applications with a different level of proactivity. One group was given applications that they would personalize to determine what the application should do for them. Another group was provided with information about how their context was changing, and the users themselves decided how to change the application behavior. The final group was evaluating applications that autonomously changed their behavior based on changing context. Additional information was also gathered from exit interviews conducted with subjects. Users indicated that they would use and prefer applications that had higher degrees of proactivity. However, as the level of proactivity increased, users had increasing feelings that they were losing control. While these findings might seem contradictory, it should be considered that owning a mobile phone constitutes some lack of control as the user can be contacted anywhere and at anytime; the user may have less control but is willing to bear this cost in exchange for a more interactive and smoother everyday experience. Beyond this issue of control, users had other concerns with regards to the usability of context-aware applications. They were concerned by the lack of feedback, or intelligibility, that the applications provided. Particularly for the more proactive versions of applications, users were unclear how they would know that the application was performing some action for them, what action was being performed, and why this action was being performed. A third concern was privacy. Users were quite concerned that the context data that was being used on mobile platforms could be used by service providers and other entities to track their location and behaviors. A final concern that users had was related to them evaluating multiple context-aware applications. With potentially multiple applications vying for a user's attention, users had concerns about information overload. Particularly when mobile and focusing on some other task, it could be quite annoying to have multiple applications on the mobile device interrupting and requesting the user's attention simultaneously or even serially.

V. Usability Risks for Mobile Context-Aware Applications

Risk that result from the use of a particular technology (in this case, context-awareness) that impact the usability of a system. In order to develop any application for user, designer has to go through all the user requirements. User and designer after discussion will have their own mental models for the application. The designer's model represents the designer's understanding and idea of the artifact being constructed, whereas the user's model is the user's conceptual model of the same artifact, its features and functionality, which has developed through her interaction with the system. According to Norman (1990), one can distinguish between the designer's mental model and the user's mental model. In order to respond to the user's needs, efficiently fulfill the users goals and satisfy the user's expectations, the designer's and user's understanding of the device or application should be consistent with each other, in other words, the user's model and designer's model should be the same (Norman, 1990).

To ensure the best possible result, the mental models of different stakeholders in application development and use have to meet each other.

i. The mental models of the application's technical designer and user interface designer should be consistent
ii. The designer's and user's mental models of the application should be the same

Without this careful design, there are significant usability risks that may result: users will be unable to explain the behavior of the context-aware application, nor predict how the system will respond given some user action.

VI. Usability Risks for Context-Aware Applications

There are several characteristics of context-awareness that can be problematic for interactive devices. Certain risks associated with developing mobile context aware applications and their consequences are summarized in below fig. 6.

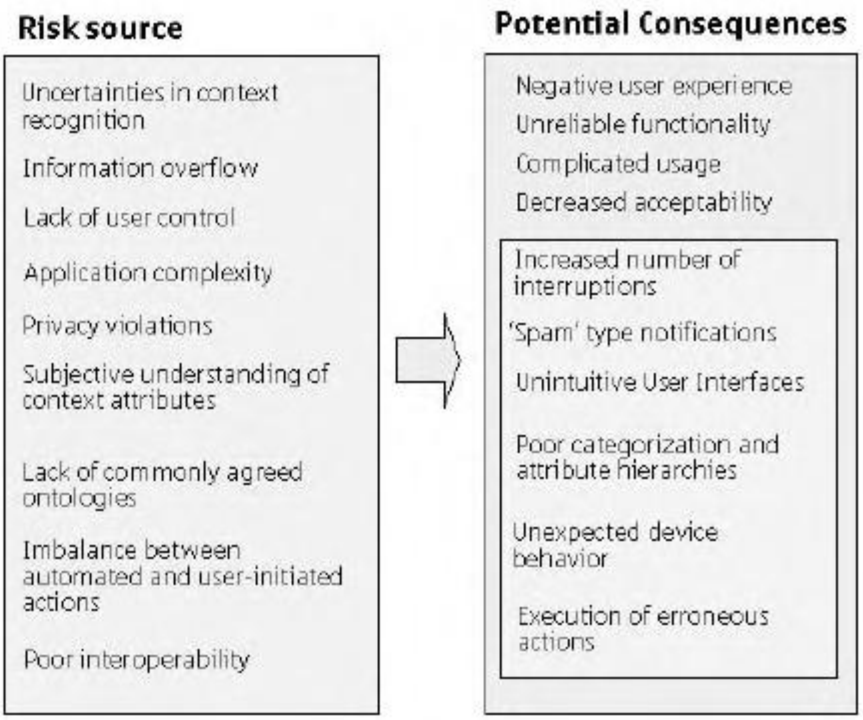

Fig. 6 Sources of usability risks their potential consequences related to context-aware mobile applications (Consequences unique to mobile applications are in smaller rectangle).

- **Uncertainty in context recognition:** Uncertainty is a part of the nature of context-aware applications. Uncertainty can be due to different reasons, such as detection accuracy, information fusion, or inferring logic. This is a key issue for designing the user interface for a mobile context-aware application, as it affects the selected features, their functionality and accuracy. In practice, features such as the proactivity level may be designed differently if the confidence level in context recognition can be estimated correctly. Thus, it is important that the application and UI designers share a common understanding of the matter and take it into account when designing both the application and its user interface.
- **Information overflow:** When personal information is shared with a number of different services, each of which will be contacting the user, information overflow often results. One can imagine a potential flow of incoming advertisements when entering a busy shopping street, if every shop within a radius of one hundred meters was to send

an advertisement to the device. Information overflow is particularly a problem for the small screens that are typical with handheld devices.

- **Lack of user control:** The lack of user control can easily occur with mobile device automation, when context-triggered actions are executed proactively. However, the promise of context-awareness is that it provides "ease of use" by taking over actions that the user does not want to do or did not think to do for them.
- **Application complexity:** It has a tendency to grow when functions are added and it forms a potential risk for context-aware applications, as they use a greater number of information sources than traditional mobile applications. Hiding the complex nature of the technology while maintaining a sufficient level of feedback and transparency so that the user can still make sense of the actions the device is performing (i.e., intelligibility) is a challenging issue. Here, the involvement of user-centric design principles is emphasized. Usability testing and user studies performed in an authentic environment combined with iterative design are key elements to producing well-performing user interface solutions.
- **Privacy violation:** Privacy violations are possible with mobile context-aware systems collecting, sharing and using a tremendous amount of personal information about a user.
- **Subjective understanding of context attributes:** Subjective understanding of context attribute creates a problem for user interface design, as the measures, such as the light intensity or noise level in everyday life are not commonly understood by end-users in terms of luxes or decibels but in relative terms such as ‘dark,' ‘bright,' ‘silent' or ‘loud.'
- **Lack of commonly agreed ontologies:** Above issue is connected to the lack of commonly agreed ontologies, which would guide the development of context-aware applications. The difficulties in categorizing context attribute and modeling context is evident from the literature (Hiltunen, Häkkilä & Tuomela, 2005; Mäntyjärvi et al., 2003).
- **Poor interoperability:** Poor interoperability of services and applications relates to the absence of standardization in this maturing field and it limits the application design, available services, and seamless interaction desired across a wide selection of devices and users. Interoperability issues have gained much attention with the current trend of mobile convergence, where different mobile devices resemble each other more and more, yet providing services for them must be performed on a case-by-case basis.

There are number of consequences resulting from these usability risks. The general outcome can be a negative user experience. This may result from an increased number of interruptions, spam, and the execution of erroneous or otherwise unintuitive device behavior. Unreliable device functionality, and unintelligible user interfaces can lead to reduced acceptability of context-aware applications in the marketplace.

VII. Design Guideline for Mobile Context-Aware Applications

Context-awareness typically contains more risks than conventional, non-context-aware technology. At the same time, context-awareness can offer much added value to the user. In order to provide this value to end-users and avoid these negative design consequences and minimize usability risks, Dey has sought to provide a set of design guidelines that can offer practical help for designers who are involved in developing context-aware mobile applications (Häkkilä & Mäntyjärvi, 2006). Below listed general guidelines have been validated in a series of user studies by Häkkilä & Mäntyjärvi, 2006 and should be taken into account when selecting the features of the application and during the overall design process.

a. Select appropriate level of automation

This is a key issue in designing user interfaces, as it affects the selected features, their functionality and accuracy. The relationship between uncertainty and selected application automation level is illustrated in below figure. As shown in

Fig. 7, uncertainties in context recognition create significant usability risks, however, by selecting an appropriate level of automation, an application designer can acknowledge this fact and address it appropriately. The greater the uncertainty is in the context-recognition, the more important it is not to automate actions. The automation level has also a direct relationship with user control, and its selection has a large impact on the number of expected interruptions the system creates for the user. The level of automation must be considered in relation to the overall application design, as it affects numerous issues in the user interface design.

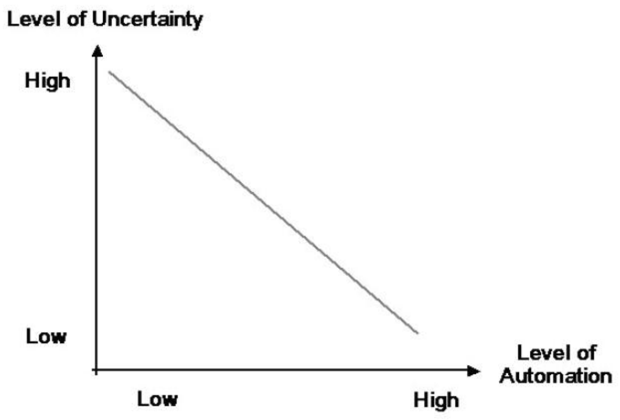

Fig. 7 How uncertainty in context-recognition should affect the selected level of automation/proactivity

b. Ensure user control

The user has to maintain the feeling that he is in the control over the device. The user, who normally has full control over his mobile device, has voluntarily given some of it back to the device in order to increase the ease of use of the device. To address this lack of user control, an important usability risk, the user must be able to take control of the device and context-aware application at any time. The desire to take control can happen in two basic circumstances- either the device is performing erroneous actions and the user wants to take a correcting action, or the user just wishes to feel in control (a feeling that users often have). The user has to have enough knowledge of the context-aware application and the device functionality in order to recognize malfunctioning behavior, at least in the case where context-recognition errors lead to critical and potentially unexpected actions. The perception of user control is diminished if the device behaves in unexpected manner or if the user has a feeling that the device is performing actions without him knowing it. User control can be implemented, for example, with confirmation dialogues however, this must be balanced with the need to minimize unnecessary interruptions, our next guideline.

c. Avoid unnecessary interruptions

Every time the user is interrupted, she is distracted from the currently active task, impacting her performance and satisfaction with the system. In most cases, the interruption leads to negative consequences, however if the system thinks that the interruption will provide high value or benefit to the user, allowing the interruption is often seen as positive. Examples of this are reminders and alarm clocks. The user's interruptibility depends on her context and the user's threshold for putting up with intrusion varies with each individual and her situation. Some context-aware functionality is so important that the user may want the application to override all other ongoing tasks. This leads to a tension between avoiding unnecessary interruptions and supporting user control

d. Avoid information overflow

The throughput of the information channel to each user is limited, and users can fully focus only on a small number of tasks at one time. In order to address the usability risk of information overflow where several different tasks or events compete for this channel, a priority ordering needs to be defined. Also, the threshold for determining the incoming event's relevancy in the context must be considered in order to avoid unnecessary interruptions. Systems should not present too much information at once, and should implement filtering techniques for to avoid messages that may appear to be spam to users. Also, information should be arranged in a meaningful manner to maintain and maximize the understandability of the system.

e. Appropriate visibility level of system status

The visibility level of what the system is doing has to be sufficient for the user to be aware of the application's actions. While this guideline has been co-opted from Nielsen and Molich's user interface heuristics (1990), it has special meaning in context-aware computing. The implicit nature of context-awareness and natural complexity of these types of applications means that users may not be aware of changes in context, system reasoning or system action. When uncertainty in context-awareness is involved, there must be greater visibility of system state in order to allow the user to recognize the risk level and possible malfunctions. Important actions or changes in context should also be made visible and easily understandable for the user, despite the fact that users may have subjective understandings of context attributes and that there may be no established ontology. System status need not be overwhelming and interrupting to the user but can be provided in an ambient or peripheral fashion, where information is dynamically made more visible as the importance value grows, and may eventually lead to an interruption event to the user if its value is high enough.

f. Personalization for individual needs

Context-awareness should allow a device or application to respond better to the individual user's personal needs. For instance, an application can implement filtering of interruptions according to the user's personal preferences. Personalization may also be used to improve the subjective understanding of context attributes. Allowing the user to name or change context attributes, such as location names or temperature limits, may contribute to better user satisfaction and ease of use. User preferences may change over time, and their representation in the application can be adjusted, for example implicitly with learning techniques or explicitly with user input settings.

g. Secure user's privacy

Privacy is a central theme with personal devices, especially with devices focused on supporting personal communication, and impacts, for example trust, frequency of use, and application acceptability. Special care should be taken with applications that employ context sharing. Privacy requirements often vary between who is requesting the information, the perceived value of the information being requested and what information is being requested, so different levels of privacy should be supported. If necessary, users should have the ability to easily specify that they wish to remain anonymous with no context shared with other entities.

h. Take into account the impact of social context

The social impact of a context-aware application taking an action must be part of the consideration in deciding whether to take the action or not. The application and its behavior reflect on users themselves. In some social contexts, certain device or user behavior may be considered awkward or even unacceptable. In such situations, there must be an appropriate balance of user-initiated and system-initiated actions. Social context has also has an effect on interruptibility. For example, an audible alert may be considered as inappropriate device behavior in some social contexts.

Once an application has been designed with these guidelines, the application must still be evaluated to ensure that the usability risks that have been identified for mobile context-aware systems have been addressed.

VIII. FUTURE DIRECTIONS

Creating context-aware interactive systems is hard. One has to keep in mind that users learn how to interact with systems, and that they adapt their behavior. It is essential that users understand the varying and adaptive behavior of the application and link it to the situations they are in. Otherwise, they will have a very difficult time learning to use the system. Hence, it is central to create understandable context-aware systems that conform to the users' anticipations. Designer may explore how appliances can be operated with minimal interaction; potentially just 'being there' is enough to work with your environment.

In short: Well-designed context-awareness is a great and powerful way to make user-friendly and enjoyable applications. If done wrong, however, context-aware applications may be a source of frustration. Just think of an automatic light, and you will probably come to think of examples that work very well, and also few others that do not. Just think of content aware app right in your garage; if the key is somewhere inside the perimeter of the car, it won't let you lock the door, or that unlocks automatically when you walk to it. But context-aware is a lot more than just location-aware.

Bulling, Roggen and Troster have mentioned in their study that the rich source of information on context that has not been used yet is the movement of eye. The dynamics of eye movements as we engage in specific activities reveal much about these activities (e.g. reading). Similarly, specific environments or locations influence our eye movements (e.g. driving a car). Finally, eye movements are strongly related to the cognitive processes of visual perception, such as attention, visual memory or learning. In addition to physical activity or location, eye movement analysis could help us infer these processes in real-world settings. Eventually this might let us extend the current notion of context with a cognitive dimension, leading to cognition-aware systems that enable novel types of user interaction not possible today. Also, Kiefer, Straub and Raubal have described impact of location as context in mobile eye-tracking studies that extend to large-scale spaces, as compared to the static eye tracking and variable contexts that characteristise natural vision.

Kristian Kersting and colleagues wrote an article 'Perception beyond the here and now' which discusses how sensor-equipped computing devices are overcoming longstanding temporal and spatial boundaries to human perceptions. It is exciting to think about how rich sensing and communication will change the way we live.

IX. SUMMARY

This study surveys the basic idea of context awareness and mobile computing. It explores better understanding of context awareness, context awareness implemented in mobile computing and various applications of context aware mobile computing. It also describes usability, risks associated with developing context aware applications and general guidelines for developing context aware applications.

Context awareness is a property of mobile devices that is defined complementarily to location awareness. Whereas location may determine how certain processes in a device operate, context may be applied more flexibly with mobile users, especially with users of smart phones. In computer science context awareness refers to the idea that computers can both sense, and react based on their environment.

There are several of development toolkits available for developing context aware application, still there are very few applications are publicly utilizing this concept. Although a lot of research has been performed in context awareness, there is much more that remains to be addressed in interacting design and usability issues for context-aware mobile applications. While developing context-aware mobile application, the understanding between user, application designer and user interface designer with each other's perspective is very important. Context aware applications offer more potential benefits to end user whilst have more usability risks than the traditional ones.

Different papers have approached these issues of usability risks and guidelines for developing applications using context aware computing; this paper combines the risks, their consequences and guidelines to overcome those risks.

In conclusion, considering all the aspects and issues of context aware mobile computing, further understanding of user requirements, their feasibility and knowledge and experience in developing context aware applications is necessary. Along with the guidelines provided in this paper, one should follow certain protocols such as security, communication, etc and proper system formats.

ACKNOWLEDGMENT

The authors are very much grateful to Department of Computer Science, Anuradha College of Engineering, Chikhali (Maharashtra) for giving opportunity to do survey work in Mobile Computing. Two authors Komal Tayde, Amit Bhala are also grateful to Dr. A. N. Nanhai, Principal of Anuradha College of Engineering, Chikhali (Maharashtra) for giving constant encouragement to do research work in the Department.

References

1. Dey AK. Understanding and using context. Pers Ubiquitous Computing. 2001;5(1):4–7.
2. Dey A. K., Häkkilä J. (2008) Context-awareness and mobile devices, in: J. Lumsden (Ed.) Handbook of Research on User Interface Design and Evaluation for Mobile Technology, Vol. I, Hershey: Information Science Reference, 205-217
3. Strang T, Linnhoff-Popien C. A Context Modeling Survey. In: First International Workshop on Advanced Context Modeling, Reasoning and Management, held as part of UBICOMP 2004, Nottingham, UK [homepage on the Internet] 2004 [cited 2013 May 2].
4. heng QZ, Benatallah B. Context-UML: a UML-based modeling language for model-driven development of context-aware web services. In: Proceedings of the International Conference on Mobile Business (ICMB'05), 2005. New York: IEEE, p.206–212.
5. Hofer T, Schwinger W, Pichler M, Leonhartsberger G, Altmann J. Context-awareness on mobile devices – the hydrogen approach. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, Hawaii, 2002 [homepage on the Internet]. 2002 [cited 2103 May 2].
6. http://www.coconutdaily.com
7. https://technology.amis.nl
8. Schmidt A. Context-aware computing: context-awareness, context-aware user interfaces, and implicit interaction. In: Soegaard M, Dam RF, editors. The Encyclopedia of Human-Computer Interaction, 2nd edition. Aarhus, Denmark: The Interaction Design Foundation; 2013
9. Krzystof W. Kolodziej, Johan Hjelm (2006, May) . Local Positioning Systems:LBS Applications and Services, p.12
10. http://web.media.mit.edu
11. Musumba GW, Nyongesa HO. Context awareness in mobile computing: A review. Int J Machine Learn Appl.2013;2(1), Art. #5, 10 pages.
12. Schmidt A, Langheinrich M, Kersting K. Perception beyond the Here and Now. IEEE Computer. 2011; 44(2):86–88.
13. Bulling A, Roggen D, Troster G. What's in the eyes for context-awareness? Pervasive Computing, IEEE. 2011; 10(2): 48–57.
14. Kiefer P, Straub F, Raubal M. (2012, March). Towards location-aware mobile eye tracking. In: Proceedings of the Symposium on Eye Tracking Research and Applications; 2012 Mar; Santa Barbara, CA. New York: ACM; 2012, p. 313–316.
15. Franchak JM, Kretch KS, Soska KC, Babcock JS, Adolph KE. Head-mounted eye-tracking of infants' natural interactions: a new method. Paper presented at Symposium on EyeTracking Research and Applications (ETRA 2010); Austin, Texas

AUTHOR(S) PROFILE

Komal Tayde, is a student of M.E. Computer Science and Engineering, Anuradha College of Engineering, Chikhali (Maharashtra). Currently she is exploring Context Awareness and Mobile Computing. She received her degree B.E. in Computer Science and Engineering in 2012 from SSGMCE,Shegaon (MH).

Prof. Amit Bhala, is Associate Professor in department of Computer Science and Engineering, Anuradha College of Engineering, Chikhali (Maharashtra) having experience in teaching in field computer science and engineering. He has done his masters in Computer Science & Engineering (M.E.).