Simulation of a Context Aware Cellphone System

Awokola J.A. (Corresponding Author)
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
E-Mail: rhgonline@gmail.com Tel: +2348069019894

Emuoyibofarhe O.J.
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
E-Mail: eojustice@gmail.com Tel: +2348033850075

Akomolafe O.P.
Department of Computer Science and Engineering,
University of Ibadan, Ibadan, Oyo State, Nigeria.
E-Mail: akomspatrick@yahoo.com Tel: +2348030848696

Amusan E.A.
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.
E-Mail: lizzydoyin@yahoo.com Tel: +2348038453840

Abstract

As the number of mobile devices we carry increase, the job of managing those devices throughout the day becomes cumbersome. This is especially true for cell phones. Despite the many benefits they provide, cell phones create problems that arise from a mismatch between the user's context and the cell phone's behavior. In large part, the mismatch occurs because owners do not remember to frequently update their cell phone configurations according to the current context. It is desirable for mobile devices to automatically configure themselves based on the context of the environment and user preferences.

While automatic configuration of cell phones may prove too tedious as different sensors are involved, it still serves as an effective means of managing cell phones in different contextual environments. In this work, a context-aware cell phone was simulated and its response was tested in different environments to see if it would react accordingly.

The sensor, media and audio manager framework of the android operating system were used in capturing data from the cellphone's inbuilt sensor, and contextual information were derived from the data captured. A combination of these contextual information was used to deduce the likely activity the user is carrying out at a particular moment, and appropriate cellphone configuration.

Keywords: Context, Context-Awareness, Cellphone, Android, Sensors, Platform

1.0 Introduction

Context is defined as the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user. The concept of context-aware computing and the definition of context were introduced in the world of ubiquitous computing in 1994 by Google research team member, Bill Schilit. Cellphones are currently the most ubiquitous communication device over the world (Reuters, 2004). Wikipedia defines Ubiquitous computing as "a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities."

Cellphones offer great accessibility and flexibility. No longer do people have to remain in a fixed location to carry on conversations over the phone. Having the ability to constantly keep in touch with people via the cellphone also gives people an additional sense of well-being (Palen *et al*, 2000). However, a lot of people already consider cell phone use in public places to be really annoying. It is not an uncommon sight to see a sign saying "Switch off Your Cellphones" in worship centers and few public places. Several governments across the world are now considering imposing etiquette on cellphone users in public spaces by legalizing technology that blocks cellular signals. Recently, Hong Kong and Canada announced they would consider legalizing jammers in an effort to curb bad cellphone behavior in public (Wylie, 2000).

The issues mentioned above can be curbed by matching the cell phone state, the context of the owner and the surrounding space, hence, a context-aware cellphone. To minimize user distraction and bad cellphone behavior, cellphones must be context-aware. By estimating user context qualities such as location, activity, physiology, schedule, and ambient context information, a human's current state can be determined to a level which allows configuring the settings of a cell phone proactively (Agre, 2001).Whenever users move from one location to another, or user's activity changes, users would have to manually adjust the cellphone profiles and settings to fit appropriately to the current situation. However, there are disadvantages associated with that process:

First, users would have to determine what phone profile would fit into the current activity being carried out which could pose as a highly daunting task and secondly, due to time constraints or undue circumstances, users would often forget to switch phone profiles before changing location or activity. This would often lead to unwanted or embarrassing disturbances in some situations. Given this, it becomes imperative to develop a cellphone system that would gather contextual information and effectively use this information gathered to decide which phone profile suits the user's current situation. This paper reports the design and simulation of a context-aware cellphone system on the android platform using an android powered device.

2.0 Context Awareness

The term context-awareness has commonly been used for two different kinds of application approaches: to capture context so that it can be later used as a cue for information retrieval, or, more commonly, to user context to adapt device behavior to correspond to the manner of its usage. In addition to these two cases, i.e. tagging context of later use and automatic execution of actions, context-awareness can be used for providing information presentation to the user (Dey, 2001). The term context has resulted numerous definitions, where none, despite of the popularity of the field, has gained a position to be considered as a default one. In the following, some well-known attempts to define context-awareness are presented. Schilit *et al.* (1994) introduced the term *context-aware computing* which *'adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time'*.

Two general categories for structuring the concept of context are proposed: human factors and physical environment (Schmidt, 2000). These have three subcategories each: human factors divide into information on the user, social environment and tasks, and the physical environment distinguishes location, infrastructure, and physical conditions. In addition, orthogonal to these categories, history provides the information on the changes of context attributes in time. Dey and Abowd (2000) define the context as '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' (Dey and Abowd, 2000). In addition they state that a context-aware application may support three kinds of features:

1. Presentation of information and services to a user,
2. Automatic execution of a service, and
3. Tagging of context to information for later retrieval.

The acquisition and processing of contextual knowledge is an area that is receiving a greater share of attention in ubiquitous and mobile research (Gellersen *et al*, 2002; Siewiorek D., 2003). Most of the focus on context aware applications is being directed towards location awareness (Peddemors, 2003). Location aware features can augment the services that are delivered by mobile phones. It is for this reason that positional context becomes an integral area in mobiles and it is further hoped that it will provide the user with a smooth transition towards context aware services (Hinckley and Horvitz, 2001; Kaasinen, 2003). The integration of the technologies required for sensing location (e.g. proximity sensor, GPS, light sensor) is becoming commonplace in modern mobiles.

2.1 Mobile Context Awareness

Employing context-awareness is not limited to mobile devices but it has been proposed or demonstrated in various different kinds of applications, for instance automated video and audio capture in a lecture room (Abowd 1999), house front doors (Kim *et al.* 2004), smart furniture (Tokuda 2004), hospital beds (Bardram 2004), or redirecting personal communication to the simultaneously most appropriate media (Nakanishi *et al.* 2000). In this section, examination is limited to mobile context aware applications, which are looked at more closely. Location is probably the most commonly used variable in context recognition, and it brings easily identified potential use cases. Location information has been used both as one contextual information source among others, and as the only context attribute, although the commonly agreed current understanding is that one should not limit context to location only.

However, location-awareness forms a significant area among mobile context-awareness research, and has great potential for future commercial applications (Kaasinen, 2003). In addition to the information about the physical location itself, the information in the means of distance and presence may provide useful data for time management and social navigation. Discount offers, information of opening hours, advertisement of forthcoming happenings, lunch menu choices, and service availability status are examples of information that may have potential value when transmitted to people passing by. Navigation aids, location-aware information delivery, and location sensitive memory aids are examples of concepts that take advantage of location-triggered device behavior. Common applications demonstrating location-awareness are tour guides in the city, campus or museum environments (Abowd *et al.,* 1997), shopping assistants (Bohnenberger *et al.* 2005), messaging systems (Rantanen *et al.* 2004), and location sensitive reminders (Dey and Abowd, 2000). Active Badge location system and ParcTab in Xerox PARC represent early work done on location-awareness. Later, location-awareness has been demonstrated in large project entities containing examples of several types of functionalities, as e.g. in Active Campus (Barkhuus and Dourish 2004) and Rotuaari project (Aalto *et al.* 2004).

2.2 Classification Of Context-Aware Applications

Pascoe (1998) presents four core features of context awareness: Contextual sensing, as the name suggests, is the services related to the current context is featured as context resource discovery. The fourth feature contextual augmentation ability to sense context information and present it to the user. Contextual adaptation is the ability to execute or modify a service automatically at run time based on the context. The ability to discover and use resources and is the ability to supplement digital data with the user's context.

Schilit (1994) classifies context aware applications based on two dimensions: the operation of the task and the way in which task is executed. Each of the two dimensions takes two values. The operation may be to retrieve information or to execute a command. The second dimension may be either automatic or manual. Applications that retrieve information for the use manually based on available context are classified as *proximate selection applications.* Applications that retrieve information automatically based on context are called automatic contextual reconfiguration. Applications that execute operations manually are called *contextual command applications*. Applications that execute commands automatically for the user based on the context are called *context triggered actions*.

Table 1: Classification of context aware applications. Source: Schilit, (1994)

Operation/Automation	Automatic	Manual
Retrieve	Automatic contextual reconfiguration	Proximate selection applications
Execute	Context triggered action	Contextual command applications

2.3 Existing Context-Aware Frameworks

Existing frameworks that support development of context aware applications include the following;

2.3.1 Java context aware framework

Java Context Aware Framework (JCAF) is the first of the kind to provide a Java based application framework. JCAF was developed to aid development of domain specific context aware applications (Bardram J. E, 2002). One of the motivations of JCAF is to have Java API (Application Programming Interface) for context awareness in much the same way JDBC is for databases and JMS is for messaging services. Figure 1 below shows the run time architecture of JCAF. JCAF infrastructure is a collection of context services connected in a peer-to-peer fashion where each context service is responsible for handling specific context information. It consists of a network of context services where each service can query the other service

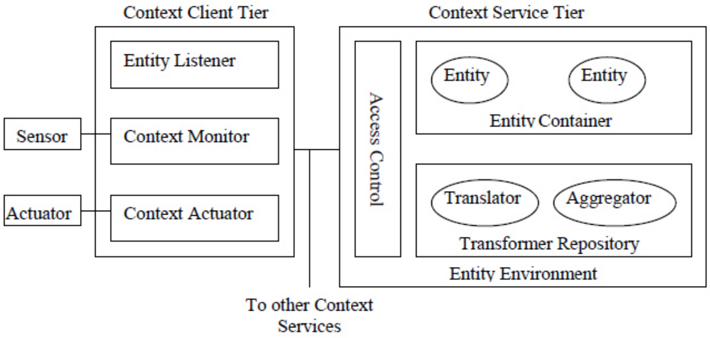

Fig. 1: Run time architecture of Java Context Aware Framework.
Source: Badram J.E, (2004)

2.3.2 Context Toolkit

The main objective behind development of context toolkit is to separate the context acquisition (process of acquiring context information) from the way it is used and delivered. Dey *et. al* (2000) uses object oriented approach and introduced three abstractions: *widgets, servers and interpreters*. Context *widgets* are software components that were introduced in order to treat context as a form of user input, to abstract the way the context is sensed, and to provide easy access to context separating the concerns and provide facility for reuse. *Interpreter* is responsible for interpreting the context by transforming the raw context data and interpreting multiple contextual data. Context Servers are associated with each entity and all the contextual information related to that entity is collected by the corresponding context (Dey and Abowd, 2000).

A number of applications have been developed using the Context Toolkit. One of the applications related to providing an interface to the user is an information display that shows the user standing in front of it, a URL related to the research group that they are in. A Dummbo application implemented using Context Toolkit uses a reader mounted to the whiteboard called Dummbo; when two users dock Dummbo, an impromptu meeting is established where the whiteboard drawings and audio are captured to facilitate the meeting.

2.3.3 Context Information Service

Pascoe, Ryan, and Morse's Context Information Service (CIS) is yet another object oriented framework which supports context aware applications. It is platform independent, globally scalable, and provides shared access to resources. Core features of CIS include contextual sensing, context adaptation, contextual resource discovery, and context augmentation (Pascoe, 1998). CIS is a layered service architecture CIS is a layered service architecture consisting of service components that include world, world archive and sensor arrays. These components are extensible and reusable. The CIS architecture consists of four CIS service components: World Archive, World, Sensor Array and a Catalog. CIS component world consists of artifacts, states, sensors, synthesizers, monitors and catalogs. Sensor arrays collect raw sensor data while synthesizers aggregate contextual data from other artifact states. Monitors control the way the contextual data are directed from sensors or synthesizers to appropriate artifacts

2.4 Generic Requirements Of Context Aware Frameworks

From the functionalities of the frameworks discussed above, a common set of requirements that any context aware framework should satisfy are listed:

1. Sensor technology to capture the contextual information: Acquire raw contextual information
2. Support for event based programming model so as to have the ability to trigger events when certain context change is observed.
3. A way to communicate the sensed contextual data to other elements in the environment and a way to interpret the collected data: provide interpreted context to application.

3.0 System Development

The process of developing the context aware system is divided into the following:

1. Identification of context information needed
2. Identification and selection of sensors.
3. Retrieval and interpretation of raw data from sensors.

4. Preprocessing of raw data to extract context information.
5. Determining user activity from context information
6. Mapping user activity to appropriate cellphone profile.

Common user activities are categorized and labeled in Table 2 below

Table 2: Categorization of common user activities

CATEGORY	ACTIVITIES
Meeting	Meeting with business partners, meeting with advisor, meeting with lecturers, meeting with students.
Studying	Reading at home, working in the lab, doing homework, research
In Transit	Driving to work, biking to school, travelling, moving from home to work/school
Relaxing	Watching TV, Taking a Nap, playing computer games
Sleeping	Sleeping

From the common user activities identified, context information needed to determine the various activities include; Location, Time, Luminance, Environment Noise Level, Speed, Device Orientation and Proximity. Having identified the context information required to develop the context aware cellphone system, various sensors that can provide the context information are identified and the suitable sensors based on power consumption rate and android framework accessibility rate are selected. Figure 2 below shows the Location acquisition model used in this work.

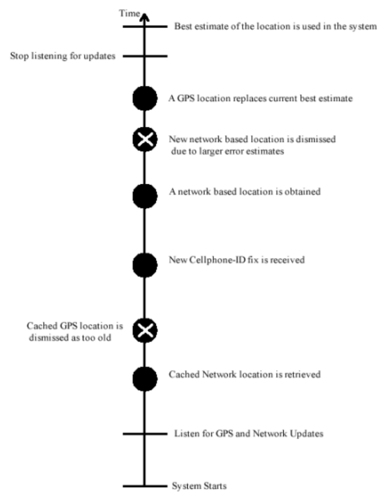

Fig. 2: The location acquisition model

3.1 Development Tools

The implementation phase of this work was carried out on the android framework. The android platform is a software stack for mobile devices that includes an operating system, middleware and key applications. The android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

3.1.1 FEATURES OF ANDROID

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WEBKIT engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)

- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plug-in for the Eclipse IDE

3.1.2 Android Framework

By providing an open development platform, Android offers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more. Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Figure 3 below shows the Android platform architecture in details.

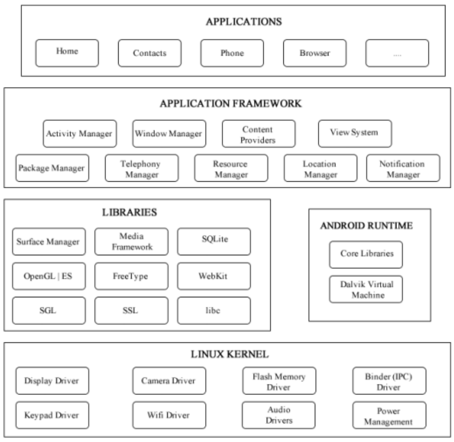

Fig. 3: The android platform architecture

4.0 Result And Discussion

4.1 The Context Aware Application

The minimum android API version targeted by the application is API 9 and was developed using the eclipse integrated development environment. The application has 3 activities namely: setup activity, main activity, about activity. The 'about activity' displays information about the context aware application. The other two activities are discussed below

4.2 Setup Activity

The setup activity allows the user to set the cellphone's home and work location. This would be allowed when the GPS system is connected as this is the location provider of the application. Depending on which location the user wants to set, the current latitude and longitude reading is set as the home/work location. Also the user is allowed to set the cellphone's default profile in the setup activity. This default profile would be set by the cellphone when the user's context reads a "normal" environment. Figures 4 to 6 below show the set up activities.

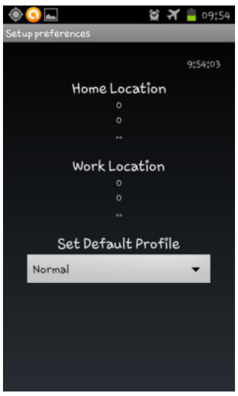

Fig 4. The Setup activity after installing the application

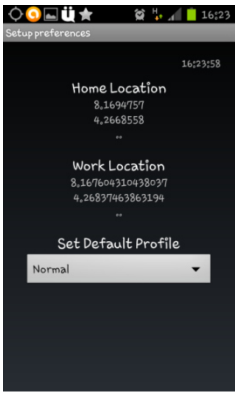

Fig 5. The Setup activity after home and work location has been set

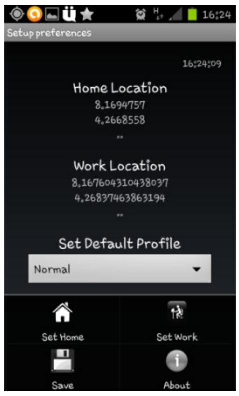

Fig 6. Setup activity and menu options available

ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol.5, No.10, 2014

4.3 Main Activities

The main activity of the application displays the interpreted readings from the sensors. Its displays the following:

i. The current user location
ii. The latitude and longitude coordinates of the current location
iii. The level of light present using bars.
iv. The level of light present characterizing it as in Table 4.3
v. The current speed of the user
vi. The orientation of the device (held or at rest)
vii. The current cellphone profile
viii. The proximity of the cellphone to any object
ix. The sound level
x. The current time of the day to seconds precision
xi. The time of the day as characterized as morning, afternoon, evening or night.

Fig 7. The main activity as the application is launched

Fig 8. The main activity prompting the user to enable GPS if disabled

Fig 9. The cellphone setting its profile to Normal

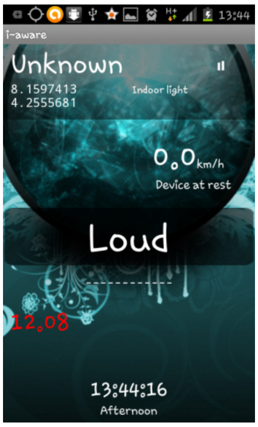

Fig 10. The cellphone setting its profile to 'Loud'

Fig 11. The cellphone setting its profile to 'Silent'

Fig 12. The cellphone setting its profile to 'Vibrate'

Figures 7 to 12 above show the main activities of the application. Figure 13 below shows the LogChat (that shows reading from the Cellphone in eclipse environment).

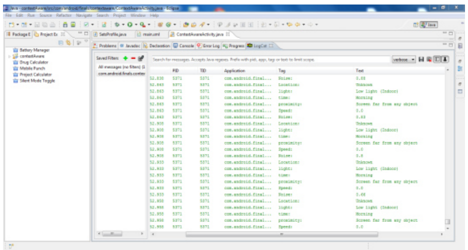

Fig 13: LogCat that shows reading from the cellphone in Eclipse environment

Figure 14 below shows the memory consumption and CPU usage of the application

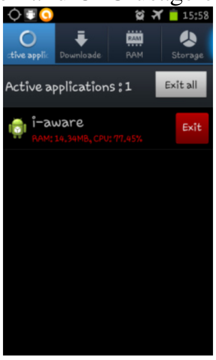

Fig 14: Memory consumption and CPU usage of the application

5.0 Conclusion and Recommendations

This paper presents the simulation of a context aware cellphone system. Previous work that has been carried on the subject was identified and the level of research reached was discussed. Frameworks used to develop context aware systems were also identified. The three major frameworks being: Java context aware framework (JCAF), Context toolkit, Context information service. The development of this work was carried out on the android platform and through the use of the Android Sensor APIs, the media framework, and the java programming language; it was shown that the android platform could support context aware systems extensively. The simulation was carried out on a SAMSUNG GT-I900 smartphone running the android v4.0.1 operating system (Ice cream sandwich OS). Also it was tested on a SAMSUNG Galaxy Pocket smartphone running the android v2.3.0 operating system (Gingerbread OS). The context aware application was tested in different environments requiring different cellphone configurations and it responded as expected.

With the mobile market share of android devices increasing to about 72.3% in 2012, the application can run on a greater share of mobile devices available. Android devices supports a wide variety of physical sensors, the android application framework uses a middleware infrastructure approach that separates context acquisition and usage and the sensor layer and the raw retrieval layer is well supported.

5.1 Recommendation

Given the constant increase in the abuse of cellphone use and in need to possess a cellphone that is truly “smart”, there arises a need to implement the functions provided by a context aware system. Hence the following recommendations are made:

i. The application developed in this work should be improved on to support learning. In the “learning” context, the cellphone is made to learn its configuration changes and time in which it changes. Having equipped with the learning function, the cellphone would be able to predict what the user would be doing, or where the user would be at a particular time. This would require the application to read data less often from its sensors and save CPU processing power, memory consumption and increase battery life of the cellphone.

ii. Current android devices, should be equipped with more sensors e.g. the barometer (for measuring pressure), the thermometer (for measuring temperature), to enable context aware application use this additional sensors to further enhance their context-sensing ability.

iii. Cellphone manufacturers should through hardware tuning and reconfigurations, make sensors built into devices consume less battery life and CPU processing power.

iv. Research should be carried out in ensuring context aware systems are plugged into every part of the human life to enable humans live life in better, convenient manner.

6.0 References

Aalto L, Göthlin N, Korhonen J & Ojala T (2004) *Bluetooth and WAP Push based location-aware mobile advertising system Proc*. Second International Conference on Mobile Systems, Applications and Services, Boston, MA, 49 - 58.

Abowd GD, Atkeson CG, Hong J, Long S, Kooper R & Pinkerton M (1997) *Cyberguide: A Mobile Context-Aware Tour Guide*. ACM Wireless Networks 3: 421-433.

Agre, P.E., *Changing places: Contexts of awareness in computing.* Human-Computer Interaction, 2001. 16(2-4): p. 177-192.

Bardram JE (2004) *Application of Context-Aware Computing in Hospital Work – Examples and Design Principles*. In Proceedings of ACM Symposium on Applied Computing: 1574 – 1579.

Bardram JE (2002) *The AWARE Architecture: Supporting Context-Mediated Social Awareness in Mobile Cooperation*. In Proceedings of CSCW'04: 192-201.

Barkhuus L & Dey A (2003) *Is Context-Awareness Taking Control Away from the User? Three Levels of Interactivity Examined*. In Proceedings of Ubicomp 2003: 159-166.

Bohnenberger T, Jacobs O & Jameson A (2005) *DTP meets user requirements: Enhancements and studies of an intelligent shopping guide*. Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE-05). Munich, Germany.

Dey, A.K. and Abowd, G.D. (2000) 'The context toolkit: aiding the development of context-aware applications', *Workshop on Software Engineering for Wearable and Pervasive Computing*, Limerick, Ireland.

Dey, A.K. and Abowd, G.D. (2000) 'Towards a better understanding of context and context-awareness', *Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness*, ACM Press, New York.

Dey AK (2001) *Understanding and Using Context*. Personal and Ubiquitous Computing 5: 4-7. Springer-Verlag London Ltd.

Gellersen HW, Schmidt A & Beigl M. *Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts*. Mobile Networks and Applications 7: 341-351. 2002

Hinckley, K. and Horvitz, E. (2001). Toward More Sensitive Mobile Phones. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology, (Orlando, Florida), 11-14 November, 2001.* New York: ACM Press. P.191-192.

Kaasinen, E. (2003). User needs for location-aware mobile services. In: Personal and Ubiquitous Computing. 7(1). p.70-79.

Kim SW, Park SH, Lee JB, Jin YK, Park H, Chung A, Choi SE & Choi WS (2004) *Sensible appliances: Applying Context-Awareness to Appliance Design*. Personal and Ubiquitous Computing 8: 184-191.

Nakanishi Y, Tsuji T, Ohyama M & Hakozaki K (2000) *Context Aware Messaging Service: A Dynamical Messaging Delivery Using Location Information and Schedule Information*. Personal Technologies 4: 221-224.

Palen, L., M. Salzman, and E. Youngs. *Going Wireless: Behavior & Practice of New Mobile Phone Users*. In Proceedings of *ACM 2000 Conference on Computer supported cooperative work (CSCW '00)*. 2000. Philadelphia, PA

Pascoe J, Ryan N, and Morse D (1998), Issues in Developing Context-Aware Computing, Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, September, pp. 208–221.

Peddemors, A.J.H., Lankhorst, M.M. and de Heer, J. (2003). Presence Location and instant messaging in a context aware application framework. In: *Proceedings of the 4th International Conference on Mobile Data Management, 21- 24 January, 2003*. London: Springer-Verlag. P.325-330.

Rantanen M, Oulasvirta A, Blom J, Tiitta S & Mäntylä M (2004) InfoRadar: Group and Public Messaging in the Mobile Context. Proceedings of NordiCHI 2004: 131-140.

Reuters, *Mobile phone users double since 2000*. Dec, 2004.

Schilit B., Adams N., and Want R. Context-aware computing applications. *First Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1994.

Schmidt, A., A. Takaluoma, and J. Mantyjarvi, *Context-Aware Telephony Over WAP. Personal and Ubiquitous Computing*, 2000. 4(4): p. 225-229.

Siewiorek, D. *SenSay: A Context-Aware Mobile Phone*. In Proceedings of *IEEE International Symposium on Wearable Computers (ISWC)*. 2003. New York, NY.

Tokuda H (2004) *Smart Furniture: A Platform for Context-Aware Embedded Ubiquitous Applications*. Keynote at EMSOFT'04 URL http://wikipedia.org/ (URL accessed on 24/11/2012).

Wylie, M., *Cell-phone jammers enforce quiet, but illegally*, in *Cleveland Plain Dealer*. 2000