Towards a User-centric and Multidisciplinary Framework for Designing Context-aware Applications

Nicholas A. Bradley
Strathclyde University
Department of Computer and Information Sciences
Glasgow, Scotland, G1 1XH, UK
Nick.Bradley@cis.strath.ac.uk

Mark D. Dunlop
Strathclyde University
Department of Computer and Information Sciences
Glasgow, Scotland, G1 1XH, UK
Mark.Dunlop@cis.strath.ac.uk

ABSTRACT
Research into context-aware computing has not sufficiently addressed human and social aspects of design. Existing design frameworks are predominantly software orientated, make little use of cross-disciplinary work, and do not provide an easily transferable structure for cross-application of design principles. To address these problems, this paper proposes a multidisciplinary and user-centred design framework, and two models of context, which derive from conceptualisations within Psychology, Linguistics, and Computer Science. In a study, our framework was found to significantly improve the performance of postgraduate students at identifying the context of the user and application, and the usability issues that arise.

Keywords
Context-aware, design framework, multidisciplinary, user-centred.

INTRODUCTION
In recent years, research into mobile context-aware computing has uncovered new ways in which to support people in their daily lives. Capabilities range from contextual sensing, adaptation, discovery, and augmentation [1], and application areas include tourist guides (e.g. [2]) and navigation systems for visually impaired people (e.g. [3]). The user's context has subsequently shifted from a traditional, static desktop environment to a dynamic, mobile setting involving a myriad of interactions with other people and objects. This transition poses many challenging, complex and largely unanswered research issues and questions.

Meyer & Rakotonirainy [4] describe how 'research into future computing technologies is often far removed from the needs of the user' and as a consequence 'the nature of such future systems is often too obtrusive'. Jang *et al* [5] state that current context-aware application development is not user-friendly. For instance, despite the insightful design principles proposed by the Ectara framework, its development involved using an artificially created scenario (involving no user studies) to test and implement six critical features of context-aware wearable and ubiquitous computing applications [6].

The neglect of usability or user-centred development is also reflective in current context-aware frameworks, which are predominantly software orientated (e.g. [7] & [8]). Bellotti & Edwards [9] highlight how such frameworks need to be expanded to handle human variability, as 'it is the human and social aspects of context which are crucial in making a context-aware system a benefit rather than a hindrance - or even worse – an annoyance'. Similarly, Dourish [10] states that 'instances of interaction between people and systems are themselves features of broader social settings, and those settings are critical to any analysis of interaction. Dourish also describes the importance of the temporal context as actions and utterances gain their meaning and intelligibility from the way in which they figure as part of a larger pattern of activity.

A shared conceptual model is also missing in context-aware design, and, consequently, there is little re-use of components or generalisations of designs. Jang *et al* [5] describe how existing frameworks do not provide a generic platform that can be adapted to different applications of context-aware computing. This may be due to the ambiguity of what is meant be context-aware computing, which has resulted in an ad hoc development and evaluation of technologies. A foundational model that facilitates the design of frameworks is therefore greatly needed.

The notion of 'context' has also not been fully captured or addressed, despite its considerable interest within other disciplines (e.g. psychology, linguistics, etc). Many researchers propound that a better understanding of context can lead to improved usability since it helps designers to decide which aspects of context to use in their applications, and which user behaviours to support [11]. Using a multidisciplinary approach to design has also recently been pushed as a required step towards more usable systems. Some researchers argue that experts in several technological domains such as software engineers, user interface experts and radio experts need to be brought together [12] in order to draw upon cognitive science, user experience and situation into the computer system design process [13].

To address these problems, this paper firstly presents our multidisciplinary model of context, which draws upon theories within psychology, linguistics, and computer science. Our model, together with a review of design frameworks for context-aware computing, is then used to propose a user-centred and multidisciplinary design framework for building context-aware systems. The aims of the framework are to:

(i) address salient human and social aspects of context-aware design,
(ii) bridge the gap between cross-disciplinary research,
(iii) integrate with, and compliment, software design frameworks,
(iv) facilitate cross-application of design principles.

The effectiveness of our design framework is also compared against another user-centred design framework, through a study involving postgraduate students. The results will be discussed in relation to previous research.

RELEVANT CONTEXT-AWARE FRAMEWORKS

In recent years, numerous design frameworks have been proposed to address the complex software and technological challenges facing context-aware computing, in particular to middleware design [e.g. 8, 14] and more generally to the process of actually building context-aware applications [e.g. 5, 6, 15], an example of which is discussed in the first sub-section. Very few frameworks are user-centred, and those that have been identified are described in the second sub-section.

Frameworks for software design

Dey, Salber & Abowd [15] propose a component-based conceptual framework for building context-aware applications. This framework was chosen for discussion since it represents a significant milestone in ubiquitous computing, and has been used as an anchor article for a special issue on context-aware computing [18].

Their approach to system architecture is concerned with actually collecting, transforming and delivering contextual information, with a focus on design reuse. The following components are discussed:

– '*Context widgets*' represent sensor abstractions that conceal details of how sensing and interpretation of the environment occurs. Widgets essentially wrap around underlying sensors and services and provide an interface to automatically deliver information to interested components or services of the system.
– 'Aggregators' store multiple pieces of low-level information (such as a person or location) that is logically related and stored in a common repository for relevant application entities.
– 'Interpreters' are responsible for abstracting low-level context data to higher-level information (e.g. using location, time of day, and travel velocity to infer user is on the train home from work).
– 'Context services' are the same as context widgets, except the output is abstracted and the actuators or change of environmental state information is controlled.
– 'Discoverers' are the final component and are responsible for maintaining a registry of what capabilities exist in the framework.

While Dey *et al*'s framework provides an insightful foundation in which context-aware applications can be built, its effectiveness has been tested on very primitive applications, namely an In/Out Board and the Context-Aware Mailing List. In these examples, only the user's location is used by the application to make an inferred decision, e.g. only mailing list members within the building receive an email. However, the extent to which this email will be of use will depend on a richer variety of contextual detail, including information regarding the user and his/her activity. Codifying this contextual detail is far more difficult, and requires more robust algorithms to manage inference, since context becomes more entangled and interrelated - an area not explored to any great detail in Dey *et al*'s framework.

Additionally, when one considers users negotiating outdoor environments involving more dynamic interactions with other people and objects, Dey's framework would need to be expanded to capture the human and social elements. These issues will also be difficult to capture using a component-based approach as designers are largely restricted to creating application code that uses existing components. Bellotti & Edwards [9] argue that people, unlike systems, make unpredictable and non-deterministic judgments about context, and so designers will need to reach beyond the application to refine or augment other components in order to deliver capabilities not anticipated by the original component builders.

Frameworks for user-centred design

Due to the large amounts of sensing required to manage unpredictable users and operating environments, Bellotti & Edwards [9] argue that there are human aspects of context that cannot be sensed or even inferred by technological means. Consequently, the authors believe that the system cannot remove human initiative. Instead, Bellotti & Edward's introduce a framework (or set of design guidelines) that 'enable users to reason for themselves about the nature of their systems and environments, empowering them to decide how best to proceed'. For this to be achieved, the notion of 'intelligibility' is introduced, where the system represents to the user what is known, how it came to be known, and what the system is doing about it. In turn, the system must enforce user 'accountability' when the context is inferred, especially when the application attempts to mediate user actions that influence other people. In order to maintain intelligibility and accountability, Bellotti & Edward's design principles are based on:

(i) informing the user of current contextual system capabilities and understandings,
(ii) providing action-outcome feedback and current and previous action feedback,
(iii) enforcing action and action disclosure,
(iv) providing user control over system and other user actions.

These design principles, and the issues raised by the authors, are a valuable, informative, and timely contribution to human and social investigations of context-aware design. However, their framework needs to be expanded to include a greater understanding of users, with respect to the decisions they make and actions they perform in different contexts. Although it is stated that users and environments are unpredictable, a systematic process for exploring spatial behaviour is not given. In order to develop more robust inference tools, application designers need to be provided with information about what is meaningful to different sets of users. This integration of the user and application's context is an important one, which is not sufficiently addressed by the authors.

In other work, Dourish [10] presents a foundation in which context-aware design frameworks can be developed, drawing on the notion of embodied interaction as developed in phenomenological philosophy. It is argued that context-aware computing needs to extend beyond the awareness of spatial location, of user identity, of the proximity of people and devices, and more towards monitoring the sociologically-motivated explorations of interaction. Dourish addresses this issue by investigating the notion of embodiment, which is about establishing meaning and relates to anything that has presence and participation in the world (real-time and real space, here and now), whether it be physical objects, conversations, or actions. Our model and framework builds on Dourish's work by providing a procedure or structure for which these design and usability issues can be captured.

PROPOSED MULTIDISCIPLINARY MODEL OF CONTEXT

Our multidisciplinary model of context, illustrated in Figure 1, is the outcome of a comprehensive review of literature and captures the relationship between different interpretations of context by researchers within psychology, computer science, and linguistics.

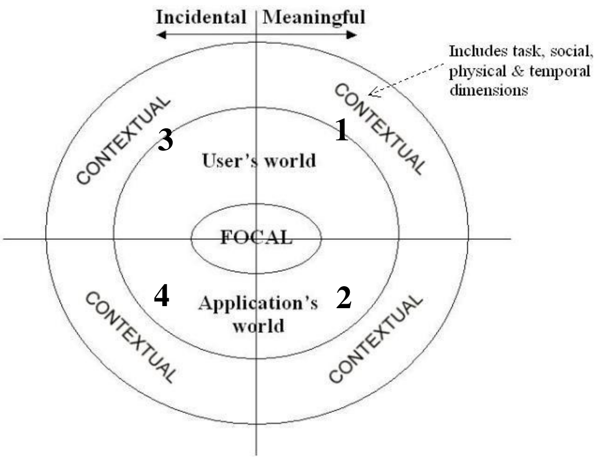

Figure 1. Our outline model of context.

The horizontal centre line, in Figure 1, separates the 'user's world' from the 'application's world'. Two major differentiations within psychology research [16] are at the heart of our model of context, and concern 'focal' versus 'contextual', and 'incidental' versus 'meaningful'.

Focal vs contextual: The oval shaped circle in the centre represents what is 'focal' to (i) the *user* with respect to carrying out actions (or embodied interactions) in an attempt to achieve goals, and (ii) the *application* with respect to transmitting information/ services to the user. 'Contextual' represents anything in the task, physical, social, and temporal contexts that influence the process with which focal user actions are undertaken and/or focal application services are executed. Table 1 describes what we mean by these contextual dimensions.

Table 1. Dimensions of context.

Dimension	Definition
Physical	The environmental location consisting of surrounding/nearby physical objects (e.g. buildings, cars, trees, etc). This also includes the presence, state and purpose of those objects, and the types of information they transmit through audio, visual, odour, texture, temperature, and movement (and in different weather conditions).
Social	The relationship with, and the density, flow, type, and behaviour of, surrounding people (e.g. sitting on a crowded train).
Task	The functional relationship of the user with other people and objects, and the benefits (e.g. resources available) or constraints (e.g. time pressure) this relationship places on the user achieving his/her goal.
Temporal	The temporal context is embedded within everything, and is what gives a *current* situation meaning, based upon *past* situations/occurrences, expected *future* events, and the higher-level temporal context relating to the time of day, week, month, or season.

Meaningful vs. incidental context: 'Meaningful' context is aspects of the environment that implicitly link to the user's primary goal, whereas 'incidental' context is aspects of the environment that just happen to be present. In order to illustrate what we mean by these terms, each quadrant will be discussed separately.

- *Quadrant 1*: The user is undertaking *meaningful focal* actions in order to realize his/her primary high-level goal (e.g. using cues and information from the environment and GPS Palmtop to navigate to the train station). The user may be influenced by, or be using, *meaningful contextual* dimensions, such as the length of time until his/her train, or information regarding the train status provided by the Palmtop.
- *Quadrant 3: Incidental* occurrences in the contextual world are normally unrelated to the user's primary high-level goal. These events may either (i) remain *incidentally contextual* if they have no impact on the user's meaningful activities (e.g. other people walking past), (ii) become *incidentally focal* if the user needs to temporarily deviate away from their meaningful activities (e.g. having to cross the street to navigate past roadwork), (iii) become *meaningfully contextual* (e.g. decide to walk another route on subsequent days), or (iv) become *meaningfully focal* if the incidental event replaces the user's current meaningful activities (e.g. deciding to visit a friend who just phoned incidentally).

- *Quadrant 2:* The application is aware of (or thinks it is aware of) the user's primary high-level goal (either inferred by application or explicitly given by user), and uses sensed data acquired from the *contextual* world to execute a *meaningfully focal* service (e.g. informing user his/her train is delayed).
- *Quadrant 4:* The application uses sensed *contextual* data, similar to Quadrant 2, to either support *incidentally focal* occurrences as described in point (ii) of Quadrant 3 (e.g. re-routing a blind user to avoid potentially hazardous excavation work), or infer possible future intentions of the user from *incidental contextual* occurrences that the user may have not noticed or has been unaware of (e.g. informing the user that one of their friends is in a nearby café).

The processes that link the contextual world to the focal world for the user and application (discussed in next section) can be interpreted through linguistics research. An utterance produced from a conversation is constructed through a task, social, physical and temporal context [17]. Instead of an utterance, a pattern of interaction between a user and computer has similarly been constructed through these dimensions of context.

Interestingly, referring back to Dey *et al*'s work, described in section 2.1, their component-based framework attempts to support very precise *meaningful* activities of the user. However, as pointed out by Bellotti & Edward's [9], it is attempting to support unpredictable or *incidental* activities that is difficult and is likely to determine whether context-aware systems are to be useful and usable.

PROPOSED USER-CENTRIC FRAMEWORK

Our design framework was constructed to address the limited (i) appreciation of human and social design issues and cross-disciplinary research, and (ii) integration of both user and application worlds. Our framework is based upon our context model illustrated in Figure 1, and is therefore divided into three sections:

- Acquisition of user context data (quadrants 1 & 3).
- Acquisition of application context data (quadrants 2 & 4)
- Usability design considerations that address the issues arising from the integration of user and application worlds.

1. Acquisition of user context data

1.1. Specify users' high-level goals and requirements.

1.2. Investigate what meaningful and incidental activities users are likely to undertake, and analyse what users use or are influenced by in the environment, drawing on (i) the dimensions of context described in section 3, namely the task, social, physical, temporal and cognitive contexts, and (ii) the notion of embodiment [10], described in section 2.2, where anything that provides a presence and participation to an activity needs to be accounted for.

2. Acquisition of application context data

2.1. List types of *meaningful* services the application could provide, in relation to 1.2.

2.1.1. For each service, indicate the types of contextual information needed to infer or identify the user's context (e.g. GPS location).

2.1.2. Explore different types of sensors, technologies, capabilities, services, and networks from which contextual information could be derived. Also consider the high level structure within which the context-aware device will function – e.g. constraints placed on the availability of contextual information.

2.1.3. Evaluate how this information might be sensed, managed, interpreted, and presented to the user.

2.1.4. Evaluate the '*application's context*' described in section 3.

2.2. In relation to 1.2, list types of *incidental* services, either to support incidental actions of the user (e.g. reminder to buy a particular book), or to infer incidental actions unbeknown to the user (e.g. informing the user of a friend in a nearby pub).

2.2.1. Repeat 2.1.1 – 2.1.4.

2.2.2. Prioritise incidental services with respect to meaningful services.

3. Usability design considerations

3.1. Investigate how meaningful and incidental application services might be presented to the user. Consider the importance of (or priority attached to) the service with respect to the user's *focal* activity and contextual environment.

3.1.1. Evaluate the timing of meaningful and incidental information (e.g. blind users may not wish incidental information when crossing a busy street).

3.1.2. Based upon 2.1.2 & 2.1.3, evaluate which output technologies should be used to provide meaningful and incidental services. Speech output, for instance, may be more appropriate when the user's task is visually demanding.

3.1.3. Investigate whether incidental and meaningful services should be pushed to the user (e.g. if his/her actions result in a dangerous situation), or pulled by the user (e.g. the user may not want his/her current activity to be interrupted) (refer to [2]).

3.2. Investigate privacy and security issues surrounding personal user information (e.g. user's location) being communicated to external sources. Which focal activities and situations would the user agree or disagree to having their location tracked by service providers, friends, family or other people?

3.3. Investigate the extent to which information should be temporally filtered once a user acquires knowledge and experience of particular contexts, activities, and situations (refer to the notion of intertextuality and co-text principles in Linguistics [19]).

3.4. If the application supports contextual augmentation, explore the human and social implications of allowing users to disseminate incidental and meaningful messages for others

Consider 3.1.3 when exploring the retrieval of messages by others, e.g. should the user leaving the message be forced to prioritise information?

3.5. Investigate how the application should respond to situations where information or services are wrongfully inferred, inaccurate, or unknown. During conflicts of interest, control should be deferred to the user [9].

APPLICATION OF FRAMEWORK

Using an example of a context-aware device to support navigation of visually impaired people, we will briefly illustrate how our framework could be applied.

User goal: To navigate independently through outdoor environments efficiently, effectively, and safely. Requirements: Information regarding hazards, traffic lights state, likely busyness of people/traffic.

Meaningful user activities: To negotiate crossings/ traffic lights, steps and kerbs, street signs, and poorly designed or maintained environmental features (e.g. potholes in the road, sloped kerbs).

Incidental user activities: To negotiate flows of people/ traffic (e.g. queues at bus stops), temporary obstacles (e.g. overhanging branches, cars parked on pavement, etc.), and excavation work on pavement.

Whilst negotiating a crossing, for instance, a blind person may use or be influenced by the contextual information in the (i) p*hysical context*: listening for car engines to indicate that cars have stopped, and feeling for tactile markings (small bumps) for alignment, (ii) *social context*: awareness of other people waiting/ crossing, (iii) *task context*: beeping from traffic lights to indicate when it is safe to cross, and the time given to cross the road, (iv) *temporal context*: dense flows of traffic/people during rush hours, and (v) *cognitive context*: some blind people prefer to wait for others to cross the street with them, and based upon past experiences, some traffic lights do not provide audio feedback.

Table 2 illustrates examples of meaningful (M) and incidental (I) services, which may provide assistance.

Table 2. Possible application services.

Service	Acquisition of information
M: State of traffic lights and countdown timer	Radio Frequency beacons positioned on traffic lights could transmit information.
M: Width of streets (2/4 lanes to cross)	User's GPS location, and detailed geographical data could be downloaded either prior to journey or in real-time through web-based servers.
I: Business of traffic and people	User's GPS location, and web-based congestion reports, derived from web cams.
I: Nearby roadwork	User's GPS location, and web-based servers revealing the locations of roadwork.

USER STUDY

In order to test the effectiveness of our framework, we compared it against Bellotti & Edward's [9], which is based upon the notion of accountability and intelligibility, as described in section 2.2.

Methodology

An introductory lecture on context-aware computing was given to 25 postgraduate students (22 males and 3 females) from the Department of Computer and Information Sciences at Strathclyde University. After, they were set a task requiring them to design *two* different context-aware systems. Students worked in groups (11 pairs, and one group of three) and to design each system they were given either (i) our design framework or Bellotti & Edward's framework, and (ii) one of two scenarios – an example of one is given below:

> *"Bob is blind and has just arrived at Glasgow Airport. He is travelling to London for a school reunion dinner. He needs to fly to Stanstead and then catch a train to King's Cross Station. This should allow him time to walk to his hotel and check in before meeting his friends in the restaurant. Bob wishes to use his context-aware device to plan for, and facilitate, his mobile activities."*

The allocation of students to each group and the distribution of scenarios and design frameworks were randomised. All groups received both frameworks and scenarios. Groups were given half an hour to design each system.

Results

The two designs provided by each group were marked by an independent examiner, who had to award marks out of 10 for each design. The score awarded to each group was dependent on how well:

- the user's context had been addressed (identification of user requirements, tasks, activities) (4 Marks),
- the application's context had been addressed (identification of useful contextual information and services, and the utlisation of different types of technologies) (3 Marks),
- usability issues have been identified and discussed (any human and social design issue) (3 Marks).

The overall results are illustrated in Figure 2, showing the differences in performance between our design framework (DeFr1) and Bellotti & Edward's (DeFr2).

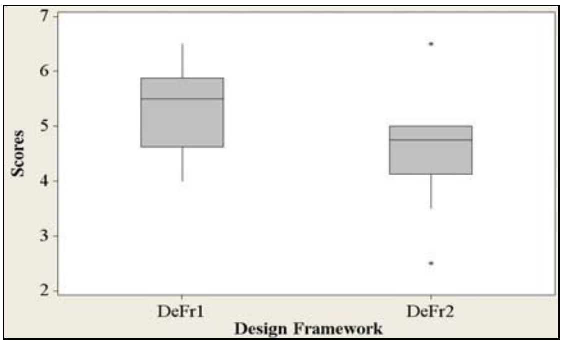

Figure 2. Overall performance using each framework.

The key findings of Figure 2, together with the statistical data in Table 3 from a two-tailed parametric related t-test, are as follows:

– When students used DeFr1 they performed significantly better overall, than when they used DeFr2 (t = 2.026>1.796, at p = 0.10). As shown in Figure 2, the median value for DeFr1 is far greater with most of the data values above the first quartile being greater than DeFr2's median value.
– Students addressed the user's and application's context significantly better using DeFr1 (t = 2.916, 2.862 >2.201, at p = 0.05).
– Students attained a higher mean value for the identification of usability issues when using DeFr2, though this result was not significant at p = 0.05.

Table 3. Testing for significance (shaded cells show a significant result.

Marking topic	Mean		St Dev		*t*-stat	Level of Sig
	Fr1	Fr2	Fr1	Fr2		
Overall	5.33	4.71	0.79	1.12	2.026	0.10
User	2.25	1.63	0.72	0.57	2.916	0.05
Application	2.08	1.54	0.70	0.40	2.862	0.05
Usability	1.08	1.54	0.70	0.62	-1.538	0.05

Post study feedback revealed that students felt pressured for time when using DeFr1. This was evident in the results, as half of the groups received <= 1 mark for the last section concerning usability issues. Consequently, this influenced the overall significance strength of the study as shown in Table 3, which explains why a significance level of 0.10 was used.

DISCUSSION

We have proposed a user-centred and multidisciplinary design framework and two models of context in order to address the following problems of existing design frameworks: (i) the limited appreciation of human and social design issues and cross-disciplinary research, and (ii) the limited integration of user and application worlds.

Integrating theories of context within linguistics, psychology and computer science, have yielded many important usability issues that have not been addressed in current research. The separation of incidental and meaningful context in psychology provides a basis in which the unpredictable nature of users and environments can be investigated, such as the response of users to incidental context. The construction of an utterance in linguistics research also enables us to investigate how different dimensions of context influence a person's embodied interactions with mobile devices, people, and other objects, and how this shapes patterns of interactions in different contexts.

Our user study illustrated that our design framework enabled students to perform statistically better in identifying the context of the user and application. By enabling application designers to address this integration, places usability at the centre of the design process. Separation of concerns in software development therefore needs to be undertaken in conjunction to human and social analyses of context, enabling the application developer to build more useful and usable context-aware systems. We feel that our context models and framework provide a valuable tool with which these issues can be conceptualised.

FUTURE PLANS

Our framework will be used to design our next user study, which will involve an investigation of usability issues of mobile navigation aids for people with visual impairments. At the centre of our research goals is to discover whether meaningful environmental information encoded by one form of visual impairment is incidental to another (e.g. loss of central vision vs. loss of peripheral vision), and whether this changes in different environmental contexts (e.g. indoor vs outdoor).

REFERENCES

1. Pascoe, J. Adding Generic Contextual Capabilities to Wearable Computers. Proc. of 2nd International Symposium on Wearable Computers, 92-99, 1998.
2. Cheverst, K., Davies, N., Mitchell, K., Friday, A. & Efstratiou, C. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. Proc. of CHI 2000, Netherlands, April 2000, pp 17-24.
3. Helal, A.S.; Moore, S.E. & Ramachandran, B. Drishti: An integrated Navigation System for Visually Impaired and Disabled. Proc. of 5th International Symposium on Wearable Computer, October 2001, Zurich, Switzerland.
4. Meyer S, Rakotonirainy A. A Survey of Research on Context-Aware Homes. WICAPUC: Australian Workshop on Wearable, Invisible, Context-aware, Ambient, Pervasive, and Ubiquitous Computing 4-7th February, 2003 Adelaide.
5. Jang, S, Kim, J, Ramakrishna, R.S. Framework for Building Mobile Context-Aware Applications. LNCS 2105, pp 139 – 150, 2001.
6. DeVaul, R.W. & Pentland, A. The Ektara Architecture: The Right Framework for Context-Aware Wearable and Ubiquitous Computing Applications. MIT Technical Report. 2000.
7. Kim, J. Yae, S. & Ramakrishna, R.S. Context-Aware Application Framework based on Open Service Gateway. *International Conference on Info-tech & Info-net 2001*, Beijing, China. pp 209-213.
8. Huang, Q. Supporting Context-Aware Computing in Ad Hoc Mobile Environments. Technical Report WUCS-02-36, Computer Science & Engineering Dep., Washington University, St. Louis, Missouri.
9. Bellotti, V. & Edwards, K. Intelligibility and Accountability: Human Considerations in Context-Aware Systems. Journal of Human-Computer Interaction 16, 2001, p. 193-212.
10. Dourish, P. Seeking a Foundation for Context-Aware Computing. Human-Computer Interaction, 16(2-3), 2001.
11. Dey, A.K. & Abowd, G.D. Towards a Better Understanding of Context and Context-Awareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, June 1999.
12. Floch, J, Hallsteinsen, Lie, A, & Myrhaug, H.I. A Reference Model for Context-Aware Mobile Services. Proc. of NIK 2001, Tromsø, Norway.

13. Selker, T. & Burleson, W. Context-Aware Design and Interaction in Computer Systems. SEWPC: Workshop on Software Engineering for Wearable and Pervasive Computing. At ICSE 2000: The 22nd International Conference on Software Engineering, 4-11th June, Limerick, Ireland.
14. Coulouris, G., Naguib, H. & Mitchell, S. Middleware support for context-aware multimedia applications. Proc. of 3rd IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems, Krakov, Poland, September 2001.
15. Dey, A.K., Salber, D. & Abowd, G.D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human Computer Interaction, 16, 2001. Special issue.
16. Smith, S. Environmental Context – Dependent Memory. Chapter 2. In: Davies, G.M. & Thomson, D.M. Memory in context; context in memory. John Wiley & Sons. Chichester. 1988. p.13-34.
17. Bunt, H. Context and Dialogue Control. Modeling and Using Context: 1st International and Interdisciplinary Conference, CONTEXT 1997, Rio de Janeiro, Brazil. 1997. p.130-149.
18. Moran, T. & Dourish, P. Introduction to Context-Aware Computing. Special Issue of Human-Computer Interaction, Volume 16, 2001. http://hci-journal.com/editorial/si-context-aware-intro.pdf
19. Connolly, J.H. Context in the Study of Human Languages and Computer Programming Languages: A Comparison. Proc. of 3rd International & Interdisciplinary Conference, CONTEXT 2001, Dundee, 27-30 July 2001. p.116 – 129.