Context Awareness Targeting User Needs

Herma van Kranenburg[1], Johan Koolwaaij[1], Ingrid Mulder[1], and Henk Eertink[1]

[1] Telematica Instituut, Brouwerijstraat 1, Enschede, the Netherlands
{Herma.vanKranenburg, Johan.Koolwaaij, Ingrid.Mulder, Henk.Eertink}@Telin.nl

Abstract. Mobile services and applications need to be able to react on changes in the end-user's context, such as available resources, user preferences, user environment and situation. Context aware support functions assist in such tailoring and should preferably not only allow applications to adapt to (predicted) changes in the user context but also anticipate user intentions and goals. This advances pro-active systems into pragmatic systems. The paper gives an overview of this vision and examples of our context aware framework and applications.

Keywords: context awareness, user centric design, pro-active responsiveness, plausible user assistance

1 Introduction

Tailoring applications to the needs and situation of mobile end-users both increases user appreciation and is a potential business enhancer. Satisfied users are likely to accept and to pay for added-value personalized applications, tailored to their needs and circumstances, always choosing the optimal communication means and serving the user with appropriate assistance, while user disappointment eventually leads to not-using such badly tailored applications. This gives the motivation to research and develop functionality that enables applications to be 'aware' of the environment they are working with. In general such functionality is coined 'context awareness'.
We follow the definition by Dey [1] for context: being any information that can be used to characterize the situation of an entity (where an entity can be a person, place, physical or computational object that is considered relevant to the interaction between a user and an application, including the user and applications themselves). In addition we consider context from a user-centric perspective. Context-aware applications use context to provide task-relevant information and/or services to a user [1,2]. Hence a context aware system provides applications with the appropriate and relevant context information that enables them to adjust optimal to the user situation.
One step beyond are ubiquitous attentive systems and pro-active systems that even enable applications to timely react on upcoming changes in context hence on beforehand changes have occurred [3]. Examples of ubiquitous attentive system responses are predicting a train-traveller will arrive in his destination-city in 5 minutes, push the bus-time schedule to him as well as information on the bus departure platform. In many pro-active systems we witness the push of potentially interesting services and applications in combination with the predicted next user location, for instance based on user preferences or application usage of users from the same age, gender, etcetera. For truly user-centric tailored applications, the attentiveness or pro-activeness should be targeted at for the user plausible and usable

results, recommendations, and contextual parameter-set predictions [4]. Such pragmatic systems support the user's intentions, his/her plausible needs and meant goals. Feedback from our context aware validators is and will be used to better understand and properly design context aware systems and in particular realize our vision of supporting users with plausible and usable results.

In this paper we will explain our vision – from reactive via pro-active towards pragmatic context aware system support. Our context management framework will be described and we will show examples – including lessons learned, from some of our context aware applications.

2 Vision

2.1 Evolution from Personal Service Environment to Ubiquitous Attentiveness

Our vision on tailored mobile applications and a user-centric system support started with the Personal Service Environment [5]. This vision, depicting a user surrounded by communication means, resources, terminals and applications nowadays is still applicable. For instance in a recently started FP6 IP [6] one of the principles is a distributed communication sphere, where a user with his terminal is considered in relation to the resources and other users within reach of him. Our PSE could be seen as a shell surrounding the end-user and taking care of discovery, negotiation, and adapting of resources and applications to the end-user and his dynamic situation. We extended the notion of a PSE into a concept coined ubiquitous attentiveness [3]. Ubiquitous attentiveness refers to the combination of 'ubiquitous computing' (i.e. lots of devices with communication capabilities), 'information systems that provide context information' and 'pro-active responsiveness' of applications. This pro-active responsiveness feature means that applications can timely react on upcoming changes in the context of the end-user, including the network and the context of any of the services that are used. A (wearable) system is ubiquitously attentive when it is cognizant and alert to meaningfully interpret the possibilities of the contemporary user space and intentionally induces some (behavioral/ system) action(s). Awareness of the contextual parameters that are relevant for the user, exchanging this information across and between (heterogeneous) domains and the pro-active responsiveness of the ubiquitous environment are thus important ingredients that are more elaborated on in [3]. An example of a ubiquitous attentive system response is predicting the next user location or next user-activity and pro-actively pushing route-assistance or entertainment services to him. An example on access network level is: while coming into reach of several Wi-Fi hotspots, instead of triggering a handover to the first sollicitated one, by taking into account additional context parameters, like travelling trajectory and predicted overlapping period with certain access network a handover to a more favorable upcoming access network will be initiated.

2.2 Pragmatic systems targeting Usable and Plausible Support

We envision that for truly user-centric tailored applications, the attentiveness or pro-activeness should be targeted at for the user plausible and usable results that not only takes into account the (predicted) changes in the (mobile) end-user environment, but also anticipates end-user's intentions and goals [4]. We called such systems pragmatic. An example illustrating this is an end-user pointing at a tv-screen, asking “brighter please”. A correct system response would be to make the screen brighter, however this would deteriorate the screen settings; while a usable system response would be to turn down the room lights or close the curtain, leading to a perceived brighter screen. Hence we expect in a pragmatic system that plausible and usable information and responses prevail over correct ones, and as such enhance and improve pro-active tailoring of applications and services.

To realize a pragmatic system, one needs to overcome many challenges. These both include understanding user intentions and goals and translating these into machine readable input as processing this input together with the contextual parameters. The processing functionality (‘reasoners’ in our context management framework, see section 3.1) in pragmatic systems will heavily rely on prediction techniques, reasoning techniques capable of processing vague and uncertain information, user behavior analysis modules, recommender and learner components [4]. Pragmatic reasoner components are needed that learn from user behavior and extend their knowledge base with learned rules for certain usage situations. Domain-specific knowledge bases are part of so-called expert systems; computer programs that provide solutions to search problems or give advice on intricate matters by making use of reasoning mechanisms. Expert systems can emulate human reasoning using appropriate knowledge representations, they can learn from past experiences by adjusting the reasoning process to follow promising tracks discovered on earlier occasions, and they can apply rules of thumb (‘heuristics’) to ‘guess’. These systems can make use of logical rules, probabilistic reasoning techniques and mechanisms that can handle uncertainty, as well as use semantic web technology. Concepts like beliefs, goals, intentions, events and situations, often including the use of ontology make these methods suitable for pragmatic systems. Proper inter-working of the behavior analysis modules, learning components and prediction engines is, of course, required.

3 Examples of context-aware applications

In this section we will describe various context-aware applications we developed in different projects. We intend to use the feedback from our context aware validators to better understand and properly design context aware systems and in particular realize our vision of supporting users with plausible and usable results.

3.1 Context Management Framework

We designed and realized a context management framework [7] that we can extend in a flexible manner. Our system comprises components taking care of sensing, collecting and processing contextual information – the Context-Source components in Fig 1– and discovering and providing access to sources of contextual information – the Context-Broker component.

messenger wrapper, which connects to Messenger software and delivers information on the status of a user on his PC or laptop and a GPS location wrapper, which is a wrapper around a Bluetooth-enabled GPS receiver connected to a Symbian phone. The role of the Context-Broker is to provide access to sources of contextual information, for which it uses a Context-Source Registry. The function of the Context-Source is to provide for relevant context and triggers by monitoring the environment. Relevant context relates to context processing challenges such as the information should be at the right semantic level, delivered at the right moment, in the requested format, conflicts resolved, etcetera. A challenge in itself is to deduce by the system (hidden) user intentions and goals; what constitutes usable and plausible and how should the system interact in a non-obtrusive way appreciated by the user.

Often the requested information is not straightforward available, and hence available information needs to be reasoned on, interpreted, combined with information from other sources, and entailed information needs to be inferred. Such derived contextual information is viewed upon as a new Context-Source which is essential in our architecture: different layers of Context-Sources can be stacked on top of each other; calling on each others interface thereby constituting a dependency hierarchy or constellation of Context-Sources.

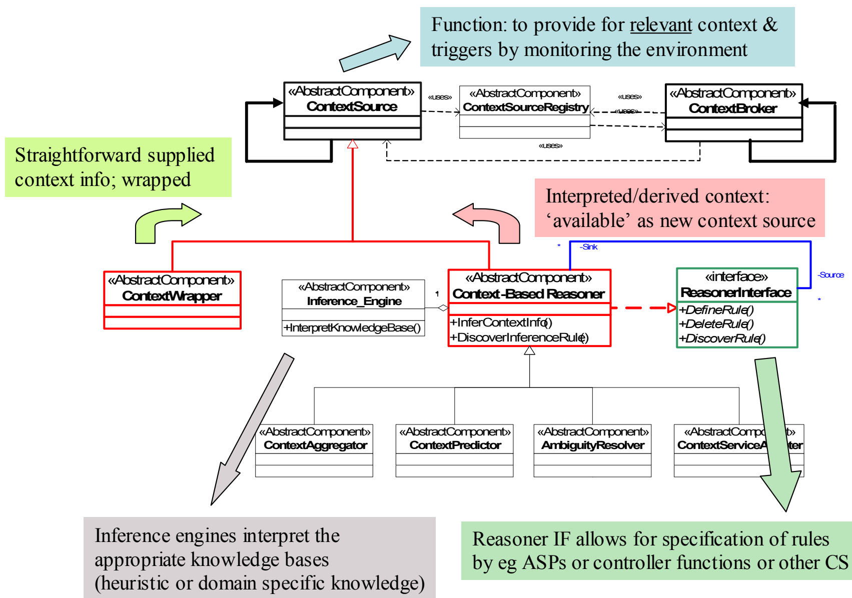

Fig 1. Functional architecture of our Context Management Framework.

In gathering contextual information we use wrapper components. For example a Reasoning bears many challenges in itself. In simple words, context reasoning is about deducing entailed contextual information from the various sources of context info, that is linked to real context data. This requires grounding; linking of your models and meta-models used in the inferring process to real context data. For the models, classical engineering methods can be used, but also several complex network modeling methods exist. The challenge is understanding which, why, when and how specific inference mechanisms compliant with multiple network contexts are used by and are valuable to people. In Fig 2 it is depicted how raw network data (e.g. from sensors) can form low-order context, while interpretations and combinations of lower-order contexts can yield higher-order contexts. Typical in low order e.g. Bayesian models used, while in higher order ontology reasoning, extended with rule-based inference can be used. In the project MobiLife (see section 3.4) this is, among others, researched upon.

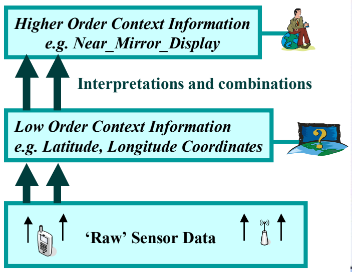

Fig 2. Reasoning schematics.

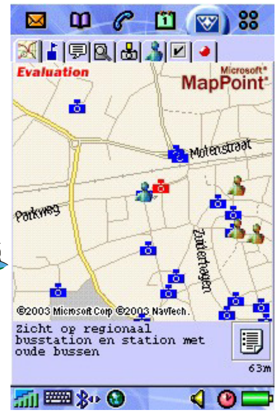

Fig 3. Compass screenshot.

Fig 4. ContextWatcher screenshot.

3.2 Compass

One of the most well-known and well-tried application domains for context-aware applications is tourism. In this domain, we have run an early pilot with the Compass application in 2003. Compass stands for context-aware mobile personal assistant, and is essentially a digital city guide that harvests touristic information from a wide range of on-line information services, based on the current location and interest of an individual tourist.

This application was built upon a context-aware service toolkit, developed in the Freeband WASP project [8]. The toolkit is tailored towards re-use of components and developer convenience and all components are exposed as web services. The screenshot in Fig. 3 shows an example where historic pictures (1880-2003) coming from different sources (monument care, municipal archives, personal collections, ...) are uniformly displayed on a map on the mobile phone, pruned based on the user's location close to the railway station and his interest in architecture. Clicking an icon results in a real-time information request to one of the services. The Compass application runs on Symbian, but advanced features require a Sony Ericsson in the P800/P910 range.

Compass is tested in a small trial in Enschede with tourists, inhabitants and experts walking different routes in the city center in early 2004. Some teams wandered through the city without a specific goal, just like tourists tend to do, other teams were instructed to simulate a young couple with child interested in city architecture or to follow a pre-defined city walk in Compass. In general, the teams were impressed by the new angle of looking at the same city that Compass provided, but at the same time it was judged as too complex for the normal tourist. This complexity was mainly in working with a PDA style phone with a stylus for the first time, the small screen with the different tabs, and unexpected delays because of the GPRS connection, but also the context-aware abilities of the Compass application necessarily cause unpredictable behaviour, which was not always appreciated by the user. However, in general the users liked the surprise effect that the system points them at unexpected places or present them with new information about known places. They also value the aspect that it is at all times possible to request information about nearby places, be it monuments or functional places such as shops or nearby toilets, and also have an easy

overview of where their friends are in the city. For local inhabitants, the ability to browse through the different periods in the history of their current location was an eye-opener, which was very much appreciated.
From a design and implementation point of view we found that a) most parties that have information about the points of interest in a city, like tourism offices and national heritage (potential service providers), do not have enough technical experience to expose their information assets via web services. Furthermore their information assets sometimes need heavy post-processing (like enriching data with latitude-longitude information or normalization of old maps) to make them suitable for application in location-based services, b) device independence is very hard to achieve. Although being our initial goal we added about 10% of native code to the generic JAVA application code to work with the phone camera and jog dial, which does not generalize to other devices, unfortunately, c) battery life becomes a limiting factor working with Bluetooth, lit screen, and GPRS continuously, and d) the platform offers functionality that is general enough to support other context-aware applications. For example we also developed a find-a-new-home application based on cell-id positioning techniques using the same platform as well as some other applications.

3.3 Abel

The experiences gained in the Compass trial were the basis for the development of Abel [9]. Abel is a digital guide for tourists who explore Twente (the region around Enschede) by bike, and it is commercially available as of April 2006. Abel basically guides tourists along predefined routes between hotels, and alerts them when points of their interest are nearby. This time we spend most of the development time on a) limiting (and not extending) the functionality for the tourist to what is really needed, b) management functionality for the commercial operator, c) making the application stable and fool proof, and d) practical issues like GPS fast fix, extending battery life time, mounting on the bikes, etcetera. Abel still has to prove itself in practice at the time of writing, but the early tests look promising. Of course, a point of concern stays the delivered content. One of the challenges of Abel is to have at least a basic but high quality content set ready before the tourists arrive in summer 2006.

3.4 Context Watcher

Context-aware life blogging is like writing your personal diaries in an automated fashion. It is no bother at all. A mobile application developed in the MobiLife project [10], named the Context Watcher automatically connects to available sensors, logs the information, detects patterns over time, and generates daily summaries about your location, activities and moods, and environmental conditions [11]. The Context Watcher is written in Python, and runs on Nokia Series 60 mobile phones (see screenshot in Fig 4). The aim of the Context Watcher is to make it easy for end-users to automatically record, store, and use context information. This can be done for personalization purposes, as input parameter for information services, or for sharing information with family, friends, and colleagues, or even just to log them for future use or to provide a mirror for the user to see his own behavior, e.g. how many times did I visit grandmother last year?

The vision behind context-aware life blogging is that man is a social being who likes to communicate about his or her experiences. The Context Watcher facilitates that process by enabling the user to submit pictures from his mobile device that are tagged with information about the context in which the picture was taken, so that title and description can be automatically generated, e.g. “I was on [business trip] together with [Henk] and [Bernd] in [Oulu] and I made this picture of the [Alexanderkatu] while traveling [with public transport] to the [summer school]”, where all the information between brackets is auto-generated. And pictures are easily discoverable by posing a simple context query like “all pictures in Amsterdam together with Bernd when it was snowing”. A second step is that the user can enable the Context Watcher to automatically generate daily summaries of his activities and send that to a (public or private) blog. Such a summary can be configured by the user and might contain an overview of all pictures of that day, the visited places, time spend there, the people you met and for how long, the weather at your location etcetera.

The Context Watcher is a thin client that handles the interaction between locally connected sensors and remote providers of context, such as for location, wellness, experiences, and photos, plus visualization and interaction in a user interface. Context reasoning and enhancement capabilities vary from enhancing e.g. bar code information to book description and personal interest profile, and reasoning to find e.g. someone’s frequently visited places based on historic location tracks and automatic tagging of the places.

The Context Watcher is running since March 2005, and has a user base of more than 150 persons of which about 50 are active each month. These users have formed many relationships, and in their hundreds of thousands context traces about 850 frequently visited places are detected. What the users liked most so far is 1) the possibility to know where the others are and to keep in touch without having to approach them directly, 2) easy access to services because the input parameters are automatically provided contextual parameters (e.g. local weather with one click, rich picture submission with one click, easy public transport info, etcetera), and 3) the sharing of context information across different (mobile) applications and web sites, including Flickr.com and their personal blogs. Sharing information via these channels with non-users of the Context Watcher (e.g. parents) was perceived as added value.

The Context Watcher application is freely available from [12].

3.5 Smart homes and your context

The Ambient Intelligence vision [13] focusses on embedded, distributed, computational power in combination with sensors and actuators in normal environments like homes, cars, and offices. This integration results in smart environments that are able to adapt their services to its current inhabitants. We are currently studying the applicability of our context management framework [see section 3.1] for these intelligent environments in the Amigo project [14]. We start from the observation that intelligent environments are able to do multiple things: trace the users, and observe the interactions between the users and the services offered by the environment (e.g. watching TV, switching on the light, listening to a particular radio station, making coffee). In an intelligent Amigo home all interactions between the users and the devices are through services, that can be automatically discovered, and have a common security model. Therefore the environment is able to monitor service usage and can derive ‘regular patterns’ in the activities of persons (e.g. grandma always goes to the bathroom first, then to the kitchen, switches on the radio, and makes some coffee). This knowledge may enable future services, such as switching on the light just before grandma enters the kitchen, or automated selection of her favorite radio channel.

In Amigo our approach is to combine intelligent environments with personal mobility [15]. For that, we defined the concept of Personal Amigo Device (PAD), see Fig. 5.. This device is able to operate as a trust-broker between your home and your current Amigo-enabled location. This PAD is seen as a guest-device in the current environment, and uses (typically) a cellular connection to its home. This scenario makes it possible for the PAD to provide detailed information (e.g. temperature, location, nearby users) towards its home environment (privacy rules permitting). Furthermore, it not only conveys contextual information, but also information about the available services. This enables one to directly use devices and services between the two environments without involving the PAD (that is only involved in the creation of the association and configuring the firewalls of both homes). In this scenario, the current context of the user includes the services of its actual environment; in other words, it closely interacts with the service-discovery mechanisms of both homes. This PAD concept enables services based on buddy lists (e.g. gaming, watching TV together) to make use of devices in the environment, instead of only personal devices. So you are able to take (virtually) part in a game with family members when in an Amigo enabled hotel-room, using the hotel TV as your interface.

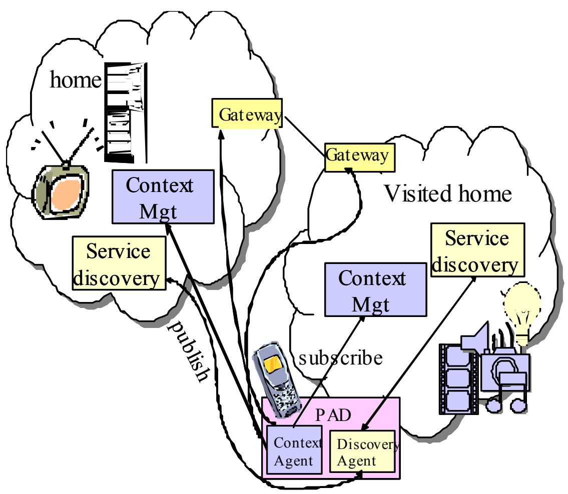

Fig 5. Personal Amigo Device scenario.

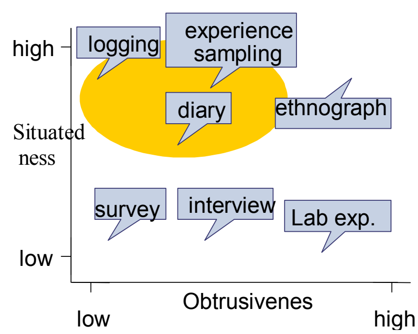

Fig 6. SocioXensor compared to other methods [18].

3.6 SocioXensor

Current research into context aware applications stresses the relevance of using context information in applications in order to improve desirable properties such as social translucence (see e.g. [16,17]). Despite occasional design successes such as Presence and Instant Messaging applications, researchers are still lacking a systematic understanding which context information is relevant in what kind of situations and which kind of applications. At the same time, designers of context-aware applications face design issues like: selecting which context information should be conveyed or aggregated to other human users (who then interpret that information), and selecting which context information is predictive enough such that it can be interpreted by applications. Although many methods exist to study social phenomena, including interviews, focus groups, surveys, laboratory experiments, ethnography, diary studies, logging and experience sampling, obtaining the right answers to design context aware applications proves to be rather complicated [18]. With our SocioXensor approach we intend to complement existing data collecting methods, compare Fig. 6. SocioXensor is a research instrument for field trials in experience and application research. It aims to strengthen logging and experience sampling by combining them with contemporary mobile and wearable devices such as smartphones and PDAs. Such devices are personal in nature, stay and travel with one person most of the time and consequently enter various contexts of that person (e.g., home, work, and mobile context).

SocioXensor is a toolkit that makes use of the hardware and software functionalities built into mobile devices for which we plan to use our context management framework. The goal is to collect information about how users experience a setting or applications. This includes aspects of functionality and usability, and emotional aspects. SocioXensor collects: objective information of human behaviour within a specific context, such as where and how communication takes place; usage information of applications, like duration of use and keystrokes usage; and subjective data, reflecting the mood of the user, like being stressed, happy, sad.

SocioXensor allows scientists to gain a much deeper and dynamic insight into the relations between user experiences, human behaviour, context, and application usage. It's results can be applied both for formative and summative evaluation, which is further elaborated on in [19].

4 Future Work and Conclusion

Context awareness can add value to applications as perceived by end-users. We consider the following key aspects of context aware pragmatic systems: awareness of the contextual parameters that are relevant for the end-user; providing the information space for modelling, storing and managing the relevant contextual parameters; exchanging the relevant contextual information across and between (heterogeneous) domains; pro-active responding and adapting of the ubiquitous environment (including the system behaviour, applications, but also sensors in the environment) on the dynamic changing context; and predicting the changes in the (mobile) end-user environment, and anticipating end-user's intentions and goals.

Lessons learned from our context aware applications include that end-users unfamiliarity with mobile terminals and mobile applications indicate that user appreciation of plausible and usable context aware support is something for the (far) future. Buddy awareness, indirect staying in touch and easy service access are appreciated by users. We continue our research and prototype validating and will use upcoming SocioXensor – and other – results for feeding our truly pro-active user support systems. Rightful understanding the end-user with his - partly hidden, likely not formulated explicitly, let alone in a machine/system readable format - intentions and goals input is likely to be a crucial factor. SocioXensor's combination of quantitative data tracked by the system (e.g., location, proximity, communication) with qualitative data provided by the users (e.g., availability, feeling, experience sampling) is expected to contribute to the acquisition of a deeper and more holistic insight of people's context, answering questions like: when designing context aware systems we make inferences on users' needs, based on contextual data implicitly sensed by the system: on which criteria should we draw a meaning out of these data? And how should the system adapt to that? Given that quantitative data are more exact, to which extent are they suitable for the understanding of the emotional aspects of experiences? Given the subjectivity of experience, what are the parameters to be assessed for different users?

Acknowledgments. This work is partly done under the Freeband projects WASP (http://wasp.freeband.nl), AWARENESS (http://awareness.freeband.nl) and FRUX (http://frux.freeband.nl), sponsored by the Dutch government and the FP6 projects MobiLife (IST-511607), Amigo (IST-004182), and SPICE (IST-027617), sponsored by the EU.

References

1 Dey, A.K.D., Abowd, G.D.: Towards a better understanding of context and context awareness". Workshop on The What, Who, Where, When, and How of Context Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computer Systems (CHI 2000), The Hague, Netherlands (April 2000)
2 Eertink, H., van Kranenburg, H., Hesselman, C.: Context-aware content distribution and adaptation, Proceedings WPMC '03, Masao Nakagawa (eds) ISSN 1347-6890 (2003)
3 van Kranenburg, H., Salden, A., Eertink, H., van Eijk, R., de Heer, J.: Ubiquitous attentiveness: enabling context-aware mobile applications and services, In: Lecture Notes in Computer Science, LNCS 2875, Ambient Intelligence, ISSN 0302-9743, Springer-Verlag, Berlin Heidelberg New York (2003) 76-87
4 van Kranenburg, H., Snoeck, N., Mulder, I.: Context aware support targeting plausible and usable results - from reactive, via pro-active to pragmatic systems, WWRF#16, Shanghai (2006)
5 Biemans, M.C.M., van Kranenburg, H., Lankhorst, M.M.: User evaluations to guide the design of an extended personal service environment for mobile services, Fifth International Symposium on Wearable Computers 2001, Zurich, Switzerland (2001)
6 http://www.ist-spice.org
7 van Kranenburg, H., Eertink, H.: Processing heterogeneous context sources, Proc. SAINT 2005, Next Generation IP-based Service Platforms for Future Mobile Systems workshop (Trento, Italy), ISBN 0-7695-2263-7 (IEEE) (2005)) 140-143
8 Koolwaaij, J.W., Strating, P.: Service Frameworks for Mobile Context-aware Applications, eChallenges workshop Future Workplaces: Supporting Mobile User and Worker (2003).
9 http://www.uitmetabel.nl
10 http://www.ist-mobilife.org/
11 Koolwaaij, J.W., et al: ContextWatcher – Sharing context information in everyday life, Int. Conference on Web Technologies Applications and Services, Calgary (2006)
12 http://www.lab.telin.nl/~koolwaaij/showcase/crf/cw.html
13 Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J-C.;Advisory Group to the European Community's Information Society Technology Programme (ISTAG): Scenarios for Ambient Intelligence in 2010 (2001)
14 http://www.amigo-project.org/
15 Ramparany, F., Euzenat, J., Broens, T., Pierson, J., Bottaro, A., Poortinga, R.: Context Management and Semantic Modelling for Ambient Intelligence, 1st International EASST-EU Workshop on Future Research Challenges for Software and Services, associated to ETAPS'06, Vienna (2006)
16 Erickson, T.; Kellogg, W.A.: Social translucence: an approach to designing systems that support social processes. Trans. on Computer-Human Interaction (TOCHI) 7 (2000) 59-83
17 Schilit, B.N.; Hilbert, D.M.; Trevor, J.: Context-aware communication, In IEEE Wireless Communications, 9(5) (2002) 46-54
18 Mulder, I., Steen, M., Mulder, I., Steen, M., ter Hofte, G.H., Kort, J.: Mixed emotions, mixed methods -- An investigation of how to study experience of we-centric context-aware adaptive mobile services [FRUX Deliverable 1.5] Enschede, Netherlands (2004)
19 Mulder, I., ter Hofte, G.H., Kort, J.: SocioXensor: Measuring user behavior and user eXperience in conteXt with mobile devices. In Noldus, L.P.J.J. et al. (eds.). Proceedings of Measuring Behavior 2005, Wageningen, Netherlands (2005) 355-358