In: Proceeding of the 8th International Conference on Human-Computer Interaction ('HCI International '99) pp. 328-332

# ATMOSPHERE: TOWARDS CONTEXT-SELECTIVE AWARENESS MECHANISMS

**Markus Rittenbruch**
Institute for Computer Science III, University of Bonn

## 1 Introduction

Within the field of computer supported cooperative work (CSCW) concepts of mutual awareness of groupware users are of increasing importance (McDaniel/ Brinck 1997). Awareness functionality supports the coordination of joint activities. The execution of cooperative work depends crucially on knowledge of the joint context of work. Common concepts of awareness often refer to the manner in which people cooperate without technical support. This is particularly relevant for the synchronous use of groupware. Systems like ORBIT (Mansfield et al. 1997) reflect, among other things, states like the presence or absence of a user and his or her working situation (busy, available, in conference, etc.).

Regarding the asynchronous use of groupware, the modeling of aspects found in non-technical cooperation seems to be more complex. While in synchronous use the context of the work which is executed jointly is often obvious, or else can be communicated by gestures or explications, the context of actions is less accessible in asynchronous use. Many of the current asynchronous awareness concepts are based on event models. User actions manipulating system objects, like documents, generate events which are recorded, stored and made available for other users interested in a specific event (e.g. Fuchs et al. 1995). On the one hand, this leads to the accumulation of a huge amount of information which has to be prepared in an comprehensible manner. On the other hand, the large amount of information makes it even more difficult to relate it to a specific working context.

This article will discuss the problem of so called context-selective awareness-mechanisms. The main issue regarding this concept is how to support users in contextualizing their actions so that these can be understood by others, even in cases of asynchronous use of groupware. In order to solve this problem, contexts have to be represented in a groupware system, and information relevant for awareness has to be related to these contexts. The Atmosphere concept described here is a lightweight approach that tries to represent the contexts of awareness information in a very straightforward manner. In particular, the concept is not intending to model real-world complexity in any way. It is rather an attempt to help users to organize shared conventions, realized as contexts, regarding the cooperative use of computer-represented artifacts. In the following, I will focus on the conceptual aspect rather than on modeling and implementational aspects of the prototype.

## 2 The notion of context

In real life a context is a complex description of shared knowledge about physical, social, historical, or other circumstances within which an action or an event occurs. In order to fully understand many actions or events, it is necessary to have access to relevant contextual information. For example, understanding the action of "opening a window" depends on whether a real window is referred to or a window on a graphical user interface.

The context of actions within distributed computer work is often not directly accessible to users. There may be situations where the context of an activity is obvious to others, especially in the synchronous use of groupware mediated by video or audio channels. In asynchronous use, however, the context within which an action took place is often hard to reconstruct. Moreover, the context of an action depends largely on the intention of the person performing that action. In order to make the context accessible to others it has to be explained, explicated or classified by the performer himself. Some of the earliest approaches to explicate the context of an action regarding mutual awareness are *informational awareness mechanisms* (Dourish/Bellotti 1992). In this model, co-workers can inform each other of their activities by means of explicit facilities, such as shared log-files. But giving a self-description for every relevant action involves a huge amount of extra work which may overburden the user and correlate with Grudin's well known 'disparity in work and benefit' problem (Grudin 1994). However, the user has to take an active part in the explanation of his intentions and of the contexts they pertain to, since these cannot be inferred from his physical actions. Therefore, tools dealing with contextualized awareness should support the user as much as possible and reduce the amount of extra work needed to create contextual information to a bare minimum. In the Atmosphere concept, we use pre-defined contexts, called spheres, which allow users or user-groups to configure contexts in advance and to select the appropriate context within which an action should take place.

## 3 The Atmosphere concept

The Atmosphere model realizes a lightweight approach to the problem of contextualization. The main objective of this approach is not to achieve completeness in the modeling of contexts, but rather to provide easy-to-use means to define basic categories of contexts.

The model is based on the abstract notion of *spheres*. Generally, spheres can be described as a kind of classification of contexts which are agreed upon by the users and within which actions manipulating particular artifacts may occur (e.g. the financial administration of project A within department B). Spheres can be configured jointly and are ordered hierarchically. Co-workers use such collectively defined spheres to focus on specific aspects of their cooperative work. The configuration of a sphere is very flexible and can represent organizational contexts (units, departments, etc.) as well as social or physical ones (time, space, etc.). A sphere contains a set of jointly used artifacts, possible sub-spheres and a set of *contextors* which represent the most fine-grained level of a particular sphere. While spheres represent particular contexts, contextors represent actions (which may be related to tasks) within this specific context. In performing an action, or a set of actions, on an artifact which is represented in a particular sphere, the performing user can choose a contextor to classify the performed action to a fine-grained level (e.g. "fixing the annual report" or "auditing of accounts"). Regarding *events* contextors and their superior spheres can be understood as cues of events.

From this follows that contexts within Atmosphere are generally represented under two different aspects. Firstly ,the decision to locate an artifact in a specific sphere is an artifact-related contextualization that relates the artifact to a context defined jointly by the users. Secondly, performing specific actions on artifacts, while selecting pre-defined contextors, generates awareness information that is located on a higher level of aggregation than basic events on artifacts (like opening, deleting etc). It is thus possible to make users aware of cues of actions rather than single events and to represent part of the intentions of the performer.

### 3.1 Design goals, architecture and implementation

The Atmosphere model is to support three major design goals: (1) to *enable the configuration of contexts*: although individual working contexts could still be useful, most of the jointly used working contexts are used by groups. Therefore, methods for defining and negotiating contexts have to be provided. (2) to *enable the selection of contexts*: assigning information about actions to working contexts involves a huge amount of extra work for the users. A pool of methods like annotations, contextors and default settings should help users to reduce extra work. (3) to *enable the presentation of contexts*: while working in different contexts, the relevant spheres have to be visible and easily manipulable. Awareness information assigned to spheres has to be displayed by selected user-defined criteria.

In order to realize these design goals, the Atmosphere model comprises three basic architectural elements. The *Sphere Browser* provides means to display and jointly modify the configuration of spheres. The *Sphere Workspace* provides means to display and manipulate the artifacts represented in spheres as well as means to contextualize actions, such as contextors, annotations, etc. The *Awareness Browser* enables users to search and browse in an awareness repository by selecting different views on contextualized events.

So far we have realized a prototype that implements several aspects of the Atmosphere model. The existing implementation is based on a Lotus Notes R4.6 platform. Since Lotus Notes is a database-oriented framework spheres are related to documents in databases that extend a specific template. The sphere browser is realized by a jointly accessible database and realizes spheres on three hierarchy levels. Spheres can be shown in different views. All events, such as modifications of artifacts, are written into a repository called event history. The awareness browser uses the event history to allow users to browse through the awareness-information and to select different views like a sphere oriented view, a view by user, a view by artifact and so on. Control of awareness-information is granted on three levels of granularity, *anonymization,* s*uppression* and *detailed access rights.*

### 3.2 Related work

A framework for activity awareness is presented in (Nomura et al. 1998). The Interlocus system, which is based on that framework, realizes individual workspaces which implement spatial frames called "regions." Due to their spatial nature, however, these may be inappropriate for larger sets of documents.

(Fitzpatrick et al. 1996) have realized a prototype based on the concept of "locales." A locale is a mapping between a social world on the one hand and site and means on the other. From this follows that artifacts may be located within locales and may have a particular context. The implementations based on this concept, such as ORBIT (Mansfield et al. 1997), are focusing on the synchronous aspects of awareness, like user presence, etc., rather than the asynchronous ones.

(Fuchs et al. 1995) introduced the notion of interest contexts. Users can define interest contexts for specific events or types of events. The notion of interest contexts was extended by (Mark et al. 1997). Work situations can be described in an informal manner to facilitate the setting of interest contexts by the use of natural language. The approach focuses on the distribution of events rather than the gathering and categorization of aggregateable structures that could represent some of the intentions of the user performing an action.

## 4 Future Work and Conclusion

Currently we are engaged in a complete re-implementation of the prototype based on the Java programming language. In this part of the project, we will focus on the implementation of contextors and the evaluation of the system in a working environment. Based on the discussion of awareness in groupware and the use of context in computer work, I have given a rough outline of a concept that aims to model aspects of working contexts in an easy manner. Several advanced aspects could not be treated here. These are, in particular, issues regarding the treatment of spheres, the administration of artifacts within several spheres, the representation of a private area within spheres, the administration of access rights on artifacts, spheres and awareness information, and other details.

I believe that the concept of contextual awareness forms a sound basis for dealing with context and intention, especially in the asynchronous use of groupware. Since it causes extra work that may not be directly beneficial to the individual, it may conflict with Grudin's 'disparity in work and benefit' problem (Grudin 1994). In this respect, I argue that there is a tradeoff between the advantage for the user and his working group on the one hand, and the extra work load for the user on the other hand. The decision as to what is appropriate may change quickly due to changing organizational or personal requirements. Consequently, it must be up to the users to decide how much contextualization they require. Further field studies will have to examine the appropriateness of the concept for different environments.

## References

Dourish, P. & Bellotti, V. (1992). Awareness and Coordination in Shared Workspaces. *Proceedings of the CSCW'92*, acm-press, 107-114.

Fitzpatrick, G., Kaplan, S. & Mansfield, T. (1996). Physical spaces, virtual places and social worlds: A study of work in the virtual. *Proceedings of the CSCW '96*, acm press, 334-343.

Fuchs, L., Pankoke-Babatz, U. & Prinz, W. (1995). Supporting Cooperative Awareness with Local Event Mechanism: The Group Desk System. *Proceedings of ECSCW'95*, Kluwer, Dordrecht, 247-262.

Grudin, J. (1994). Groupware and social dynamics: eight challenges for developers. *Communications of the ACM* 37(1), 92-105.

Mansfield, T., Kaplan, S., Fitzpatrick, G., Phelps, T., Fitzpatrick, M. & Taylor, R (1997). Evolving Orbit : a progress report on building locales. *Proceedings of the Group 97*, 241-250.

Mark, G., Fuchs, L. & Sohlenkamp, M. (1997). Supporting Groupware Conventions through Contextual Awareness. *Proceedings of the ECSCW '97*, Kluwer Academic Publishers., 253-268.

Nomura, T., Hayashi, K., Hazama, T. & Gudmundson, S. (1998). Interlocus: Workspace Configuration Mechanisms for Activity Awareness. *Proceedings of the CSCW '98*, acm press, 19-28.