Computer-Supported Cooperative Work Group
Faculty of Media
Bauhaus-University Weimar

Technical Report #: BUW-CSCW-2006-04 April 2006

COBRA: A Constraint-Based Awareness Management Framework

Tom Gross, Christoph Oemig
Faculty of Media, Bauhaus-University Weimar, Germany
({tom.gross, christoph.oemig}(at)medien.uni-weimar.de)

Contact:
Prof. Dr. Tom Gross
Faculty of Media
Bauhaus-University Weimar
Bauhausstr. 11, Room 113
99423 Weimar, Germany

E: tom.gross@medien.uni-weimar.de
T.: (++49-3643) 58-3733
F: (++49-3643) 58-3709
W: http://www.uni-weimar.de/medien/cscw/

Gross, T. and Oemig, C. *COBRA: A Constraint-Based Awareness Management Framework*. Technical Report # BUW-CSCW-2006-04, Computer-Supported Cooperative Work Group, Faculty of Media, Bauhaus-University Weimar, Bauhausstr. 11, 99423 Weimar, Germany, Apr. 2006.

COBRA: A Constraint-Based Awareness Management Framework

Tom Gross, Christoph Oemig
Faculty of Media, Bauhaus-University Weimar, Germany
({tom.gross, christoph.oemig}(at)medien.uni-weimar.de

Abstract. The effective and efficient cooperation in communities and groups requires that the members of the community or group have adequate information about each other and the environment. In this paper, we outline the basic challenges of managing awareness information. We analyse the management of awareness information in face-to-face situations, and discuss challenges and requirements for the support of awareness management in distributed settings. Finally, after taking a look at related work, we present a simple, yet powerful framework for awareness management based on constraint pattern named COBRA.

1 Introduction

The effective and efficient cooperation in communities and groups requires that the members of the community or group have adequate information about each other and the environment. This information can, for instance, include data about the presence and availability of others, as well as data about the state of shared artefacts. In the fields of Computer-Supported Cooperative Work, Human-Computer Interaction, and Ubiquitous Computing a great number of concepts, systems and prototypes have been presented to support awareness emphasising its importance [e.g., Beaudouin-Lafon & Karsenty 1992; Dourish & Bellotti 1992; Gutwin & Greenberg 2002; Sohlenkamp 1999]. However, several challenges still remain [cf. Schmidt 2002]. In particular, the so-called dual trade-off between awareness and disturbance and awareness and privacy [Hudson & Smith 1996] has stayed unsolved issue: On the one hand the amount of incoming information originating from the activities of others is overwhelming. On the other hand privacy concerns demand for regulations regarding the information being sent to others about oneself. Yet, all this information is needed to create and maintain awareness providing a basis or context for our own activities [Dourish & Bellotti 1992].

While dealing with these issues implicitly in face-to-face situations, a facility managing awareness information is needed in the digital realm for distributed communities and groups. Several suggestions analysing the structure of awareness information [e.g., Gutwin & Greenberg 2002; Sohlenkamp 1999], and for handling awareness information [Fuchs 1998; Sohlenkamp 1999] have been made. However, they mostly provide incomplete solutions.

In this paper we introduce COBRA—a COnstraint-Based awaReness mAnagement framework—COBRA. We start by discussing the basic challenges of managing awareness information. We analyse the management of awareness information in face-to-face situations, and discuss challenges and requirements for the support of awareness management in distributed settings. The following sections present a simple, but yet powerful framework for managing awareness, which is based on constraint pattern. We show how social and physical constraints can be modelled in the framework and provide recommendations for the implementation of this framework with a design pattern. Finally, we compare this framework with other related approaches and systems.

2 Managing Awareness

We define awareness management as the process of controlling incoming awareness information of others (thus the control of disturbance) and outgoing awareness information about oneself (thus the control of one's privacy) in digital communication environments. Initially the term awareness management has been used in the context of collaborative virtual environments (CVE) [e.g., Antunes *et al.* 2001]. However, according to Antunes et al., the main goal of awareness management is to reduce information overload in letting the user only process the information relevant to him—thus considering only incoming information. The overall issues addressed by awareness management in CVEs are information overload, efficiency, system performance and scalability. However, aspects regarding outgoing information are missing entirely.

Managing awareness is similar to basic activities in the area of information filtering: Malone *et al.* [1987] distinguish positive and negative information filtering techniques. They refer to selection as a positive way of filtering which picks things from a larger set of possibilities. A complementing negative way of filtering is the removal of certain elements from a given set.

Realising the need to filter information raises the questions of where and how to apply what kind of filters. Taking a closer look at awareness , a very prominent and early definition in the area of CSCW comes from Dourish and Bellotti [1992]: „Awareness is an understanding of the activities of others, which provides a context for your own activity" (p.107). Yet, from an awareness management perspective this notion appears not to be sufficiently precise; in particular, it leaves unclear how we can manage it.

The term awareness appears not only to be intuitively defined in CSCW [cf. Sohlenkamp 1999] but also very problematic itself [cf. Schmidt 2002].Therefore, as Gutwin and Greenberg [2002] we want to utilise an awareness conception derived from the area of aviation psychology, which describes it as a mental model serving as a common basis between perception, comprehension, and projection processes [Endsley 1988]. This allows conceiving awareness in two ways [Gutwin & Greenberg 2002]:

- Awareness just as mental model (narrower sense).
- Awareness as mental model including its information acquisition and exposition processes (wider sense).

Whereas it is more difficult to influence awareness in the narrower sense, it is easier to influence awareness in the wider sense (e.g., by providing certain cues—that is, carefully selected pieces of information [cf. Bogdan & Sundblad 1999]).

3 Requirements

In order to learn more about how to manage awareness we first analyse the management of awareness in face-to-face situations, and then discuss the management of awareness in the digital realm deriving requirements for awareness management implementations.

3.1 Face-to-Face Situations

In face-to-face situations humans have natural behaviour and conventions for both creating their own awareness, as well as stimulating other's awareness about oneself.

For creating and maintaining awareness, the continuous perception of plenty of information is necessary. One's environment can also be considered as a certain amount of information it may generate. Nevertheless, the actually perceived environment is limited by individual perceptive and cognitive capabilities. Thus awareness has an inherent limitation which technology is not able to overcome. In the real world we are used to shifting our attention by focussing on certain things while neglecting others. We do not perceive everything at the same time the same way—that is, we control our information acquisition and maintenance processes, mostly implicitly.

In the case of information exposition (i.e., actions or behaviour, by which awareness of other's about oneself is stimulated) humans rely on inferences they make based on their mental model. A key factor determining what to expose is the current situation or context. One of Merriam-Webster's context definitions describes it as "the interrelated conditions in which something exists or occurs" [Merriam-Webster 2004]. Yet, to relate to something else (especially to other people), one also needs to be aware of oneself. Depending on how we perceive ourselves in relation to other people we expose certain aspects of our identity: "One's identity is comprised of both a personal internal identity and a public social identity. As people engage socially, they project aspects of their internal identity into a social identity for others to perceive. Based on the situation, people only present a particular facet of their internal identity for consideration" [Boyd 2002, p.11].

According to Goffman [1956] one reveals a face, which corresponds to a social presentation of a facet of one's identity. People may maintain not just one exposable face but several ones—often associated with a particular context, especially situational and interpersonal contexts. The former is comprised of environmental factors like time, location, occasion, politics and general social values while the latter is affected by the current set of people and their relationship (roles, identities, commonalties) to the individual (cf. Figure 1).

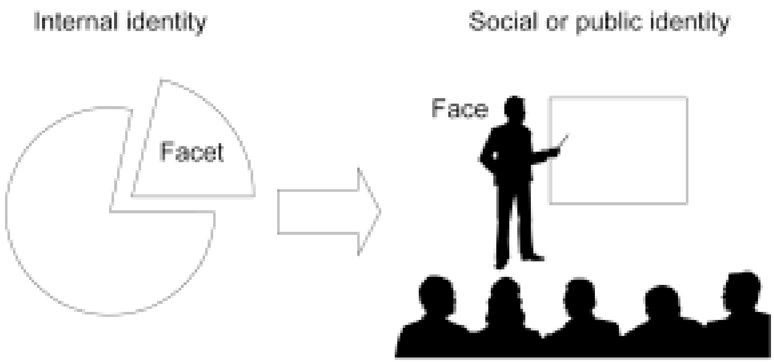

Figure 1. Facets become faces.

3.2 Distributed Situations

With the users' interaction being mediated by (computer) technology, acquisition and exposition of awareness information cannot be done any longer in familiar and intuitive ways. This entails three major problems:

- Opposed to the real world nearly every awareness activity becomes an explicit effort—due to the user's disembodiment. According to Goffman's performance theory the passage of social information is comprised of three components: information, which is explicitly given, information, which is unconsciously given off, and inferences made by the observer [Goffman 1956].
- There might be data that cannot be captured or presented at all, which means a loss of information. For instance, subtle nuances, which can only be captured by humans, who are co-present at the same place.
- Due to explicit information provision a user becomes overwhelmed a lot faster by his digital device than in the real world. Yet, means dealing with the user's perceptive capabilities more efficiently are usually missing.

Since the user is not immersed in his environment and information is captured for him, he has to rely on the application developer's expertise to provide sufficient means to perceive the environment properly and to be able to expose the appropriate information (or facet) at the appropriate time. Thus the developer carries a great responsibility: Selecting what to capture and to present and how to present it tremendously impacts a mental model's content and structure [Sasse 1997], which in turn influences the user's inferences and projections. Selecting too much information will overwhelm and disturb the user, selecting too little or the wrong information yields an erroneous or insufficient mental model. A socially inappropriate choice of awareness information and its gathering may easily raise privacy concerns.

Yet, besides negative aspects the digital realm offers also new possibilities, which have to be considered in managing awareness as well: Completely different contexts may occur concurrently (e.g., one may be chatting with two people belonging to totally different interpersonal contexts). The digital world also allows the storage of data thus imposing the need to manage synchronous and asynchronous awareness information. However, this fact causes another major problem at the same time: It lets context collapse with ease [Boyd

2002]. Information exposed in a certain context may be retrieved under totally different circumstances with unpredictable effects causing the user a great discomfort.

These problems and challenges lead to the following major design requirements:

1. Incoming and outgoing information has to be considered in order to address the two aspects of the dual trade-off (privacy and disturbance). We especially seek the selective dissemination of information and context-dependent reception.
2. Due to the user's limited perception technology needs to focus on efficiency. As its real world counterpart managing awareness in digital systems has to be as effortless as possible, thus staying a secondary task. A facility has to offer efficient means to controlling incoming and outgoing information.
3. Information selection has to be done very careful preferably based on context. Especially supporting situational and interpersonal context has to be a key goal. It is not enough to merely capture information but also its context of origin and to determine the recipient's current context of work [Gross & Prinz 2003].
4. In order to support interpersonal context the user must be provided with means supporting self-awareness such as offering views from another user's perspective (corresponds to Erickson's [2003] third-person view).
5. An awareness management facility has to work in synchronous and asynchronous modes, that is, in scenarios of blended synchrony. Context needs to be preserved in both cases.
6. Concurrent contexts need to be supported but their separation has to be ensured.

These design challenges were central issues in developing our simple constraint-based awareness management framework introduced below.

4 A Constraint-based Awareness Management Framework

Based on the design requirements outlined above, we developed COBRA—a COnstraint-Based awaReness mAnagement framework—which is comprised of a four-staged process chain applying a constraint pattern as information filter. We describe the information flow in COBRA, and then explain filters as a means to adapt the information to the respective situation. Social and physical constraints are then used to exemplify filters.

4.1 Process Chain

As mentioned earlier, in the digital realm we are only able to manipulate the processes surrounding the user's mental model (awareness in its wider sense). Roughly, information needs to be gathered from one user or particular situation, distributed and eventually presented to another user or another situation. Furthermore, in an additional stage—the process stage—information is enriched by external data sources (e.g., LDAP or other databases). Additionally, information may also be manipulated to comply with certain formats or conventions, which means that some of its parts may be rearranged or even be removed again. Figure 2 show the four stages.

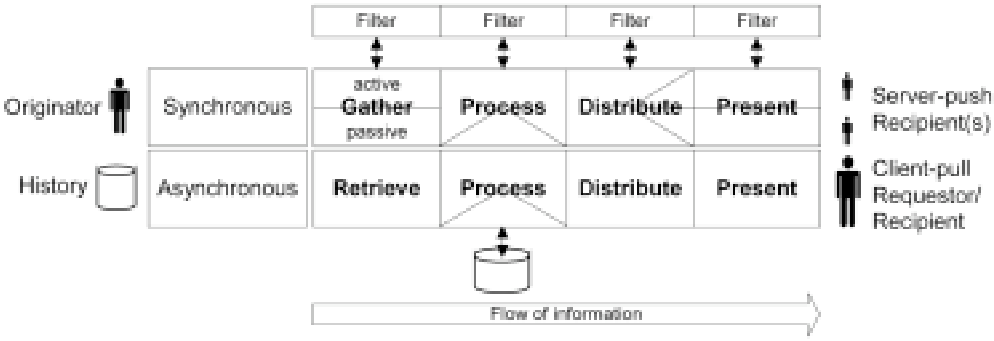

Figure 2. A four staged process chain.

It offers asynchronous and synchronous modes of dealing with awareness information. Both can be distinguished by the information's source, the way it is disseminated (server push or client pull), and by the number of recipients. The synchronous gathering stage considers information that can be either actively generated or passively collected from its originator [cf. Bogdan & Sundblad 1999].

4.2 Filtering

Constraints represent circumstances and conditions that have an impact on our interaction by supporting or inhibiting specific behaviour. They are restrictions on our freedom to pursue certain activities [Buergy & Garrett 2002]. In this framework situations can be described by a constraint pattern—that is, a specific set comprised of constraints from various categories [Buergy & Garrett 2002]. Thus constraint patterns may be used as context descriptions. At each of the four stages constraint-based filters can be applied.

There are several kinds of constraints, which can be grouped into constraint categories. We suggest the following set of categories:

- Physical constraints (e.g., proximity, size, loudness, acoustic)
- Social constraints (e.g., politeness, moral, decency, norms, relations and roles)
- Individual constraints (e.g., cognitive, logical, physical, and physiological capabilities as well as expertise and experience)
- Legal constraints (e.g., rules, regulations, and laws)
- Organisational constraints (e.g., organisation specific rules and regulations)
- Technical constraints (e.g., limitations due to the equipment used)

While there are constraints that can be violated (e.g., social constraints), there are also others that cannot (e.g., physical constraints). Some are harder to implement than others, some even ought to be left to mechanisms outside the digital realm like social protocols [Preece *et al.* 2002].

It is possible to take more than one constraint from one category to form a constraint pattern (e.g., implementing hearing and seeing ranges). However, one and the same constraint pattern ought to be applied to all process stages. Switching patterns across stages is not appropriate to describe a common context for users. Applying more than one constraint at a process stage might imply a general sequence. Yet, this depends on the constraints involved. Figure 3 denotes the final structure of our constraint-based awareness management framework.

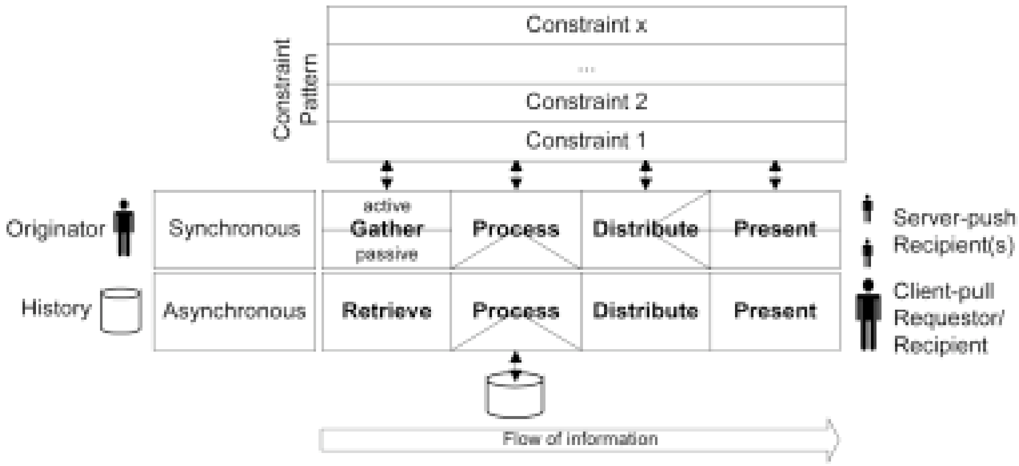

Figure 3. A constraint-based awareness management framework.

To meet further design challenges, we created a constraint pattern comprised of a social and a physical constraint.

4.3 Facets as Social Constraint

Facets expose certain information to a group of people. Thus they correspond to Goffman's facets described earlier. They help to set up interpersonal contexts. The social constraint is realised as facets similar as in Boyd's [2002] SecureId, which was inspired by Viegas' [1997] work on "Collections".

In our framework a user may define multiple facets, each with a name, which may be opened or closed determining the sending and reception mode of information associated with a particular facet. Thus, a user could have a particular facet with working colleagues (e.g., allow them to see the working email address), and a different facet with private friends (e.g., allowing them to see the private email address as well as the private mobile phone number). One or more open facets allow for concurrent contexts. An active facet promotes the user's current context. Every user has a public facet, which corresponds to a general face exposed especially to new or unknown contacts.

Two users are related to one another via one or more bridges. A bridge describes a user-facet combination in relation to another user-facet combination. For instance, user A assigned user B to his facet 1, while user B assigned user A to his facet 2 (i.e., A-1:B-2). This reveals that facets are not necessarily symmetrical but personal—as in the real world. User A might be in user B's "work" facet, while the user B is placed in A's "public" facet. There may be multiple bridges describing a relationship between two users. Additionally, a bridge allows determining when another user is within or outside one's context (the one originally negotiated and stored as a bridge).

Figure 4 depicts a simple scenario: User A has three facets (two opened, one of them active, one closed). User B has two facets (both opened, one of them active). User A can infer from user B's representation (dotted line) that user B is currently in another context than negotiated by their bridge.

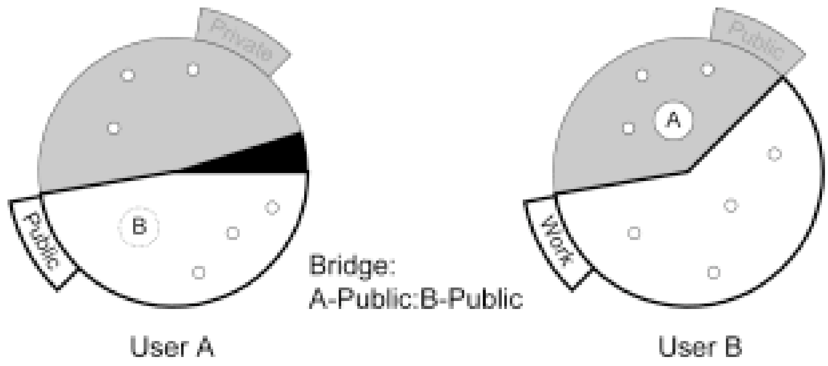

Figure 4. Facets and bridges: User B being in a different context than negotiated by the bridge.

4.4 Aura, Focus, and Nimbus as Physical Constraint

The concepts of aura, focus and nimbus were introduced in the spatial model [Benford & Fahlen 1993]. Basically, the aura is the area, where interaction may occur; the focus is the area of the user's attention; and the nimbus (or emissive focus) is the area where information about the user is exposed.

In our framework they are used as physical constraint. Each facet represents an aura—i.e. an area where interaction may occur. It is subdivided into nimbus and focus. The nimbus is the area where information about the user is provided while the focus represents the area of the user's attention. Here, objects may appear larger than inside the nimbus.

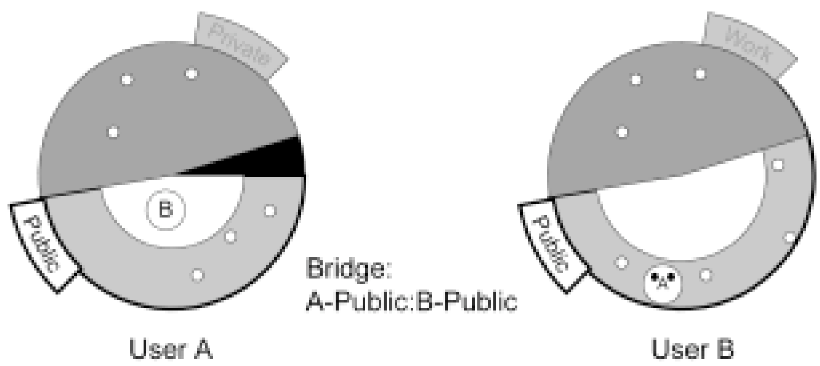

Figure 5. Focus and nimbus in an active facet. User B is in focus of user A.

The level of awareness on both sides depends on a combination of the users' foci and nimbi. Yet, only active facets have a focus area. Focus and nimbus can also be distinguished by the degree of details they provide. For instance, having another user in one's nimbus reveals lightweight information needed for making contact. Activity may be indicated by movements of his representation, inactivity by its drifting to the periphery of one's nimbus (as in social proxies [Erickson & Kellogg 2000] but here also outside a conversation). Dragging another user's representation into one's focus may yield more detailed information about his activity (e.g., that he is talking to someone else or his screensaver has been on for the past ten minutes). While facets reveal information about current contexts, further information can now be provided concerning oneself being in another user's focus or nimbus. Figure 5 illustrates user B being in user A's focus.

By opening or closing further facets, one may increase or decrease the level of possible disruption. At the same time a facet communicates a certain context allowing selective dissemination and context-dependent reception of information. A distinction of focus and nimbus offers more efficiency reducing the chance of being overwhelmed.

4.5 Discussion

COBRA provides answers to all of the above mentioned requirements. It does not only consider incoming and outgoing information but especially regards disturbance and privacy using facets, focus, and nimbus. These means allow controlling the flow of information by opening, closing, and activating facets and focusing on certain things. Although still requiring explicit activity they reduce the effort to natural and thus familiar ways. Thus COBRA tries to be as efficient as possible allowing selective dissemination and context dependent reception of awareness information. Situational and interpersonal contexts are supported by facets and focus. Bridges relating two user-facet combinations express the context of origin and the current user's context of work. Selecting another facet view a user may find out what information he exposes to other facet-members. Additionally the facet name (e.g. "work") expressing his current context may help to become more aware of himself in relation to others.

Blended synchrony is supported by default due to the underlying process chain. Providing facet and user based context separation concurrent contexts do not become a problematic issue.

Overall, the concept of COBRA fulfills all of the requirements with the little exception of the remaining effort, which is mainly due to current human-computer interfaces.

5 Implementation and Integration

In this section the implementation and integration of the concepts above is described. First, we introduce the mediator pattern, which is used. Then, we present the application of this pattern in the implementation.

5.1 COBRA Mediators

The above concept of COBRA is realised as a mediator pattern, which is widely used in object-oriented modelling of software systems [Gamma *et al.* 1994]. It is comprised of a central controlling instance (called mediator in the pattern language), which promotes loose coupling by keeping the collaborating objects (called colleagues in the pattern language) from referring to each other (cf. Figure 6).

The mediator controls and coordinates all ongoing interaction and represents the application's overall behaviour. All communication between colleagues goes through the mediator—colleagues cannot refer to each other directly. The mediator receives and forwards information to the respective colleague(s). In the COBRA framework the communication works according to negotiated bridges. The mediator also realises the connection to other external services needed at the process stage like databases or directory services. Additionally, the mediator has knowledge of the whole situation and all bridges to its colleagues and external data sources. Therefore, the mediator can also be used as a basis for the context-sensitive social network analysis. For instance, a graphical representation of the social network among the users could be generated automatically.

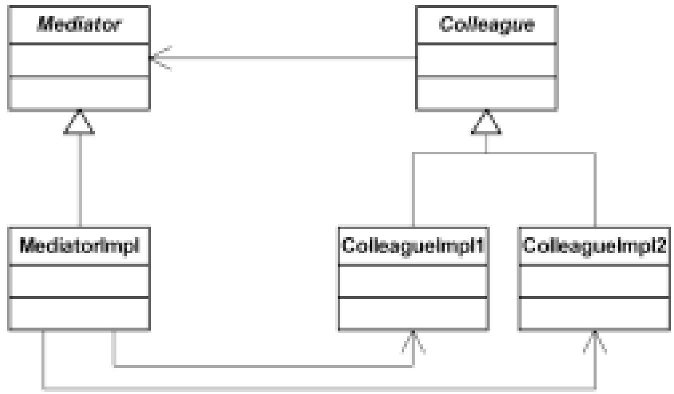

Figure 6. An UML diagram of the mediator design pattern.

A colleague is an entity, which can gather information from its environment, or present information to its environment. There are three types of colleagues:

- Colleagues just gathering information. These are mostly so called agents or bots, which constantly provide information about a certain topics (e.g., the stock market).
- Colleagues just presenting information. These can be used to integrate awareness features into other applications. For instance, a colleague of this type could present a user’s status on a Web site or in an expertise location system. Only one facet needs to be bridged to this colleague, thus the user can control his availability in those systems with the same mechanisms described above as for any other system or user.
- Colleagues gathering and presenting information. These are regular clients as described in previous sections.

All colleague types may be combined in facets with the user being able to control the information they send us and the information they receive from us.

On this general level of the pattern any common information format can be used for the communication among the mediator and the connected colleagues. Any information has to be represented in this common information format—that is, information in other formats (such as data from external sources) has to be adapted in contents and format to the respective common format.

5.2 Application of the COBRA Mediator

The COBRA framework and its mediator pattern are implemented as a client-server infrastructure. There has to be a central mediator, which is always running, in order to guarantee persistency. This is particularly the case for asynchronous situations, where some colleagues simply gather information and it is not yet clear how this information will be used in future. Figure 7 shows an instantiation of the implementation of the COBRA framework as client-server architecture.

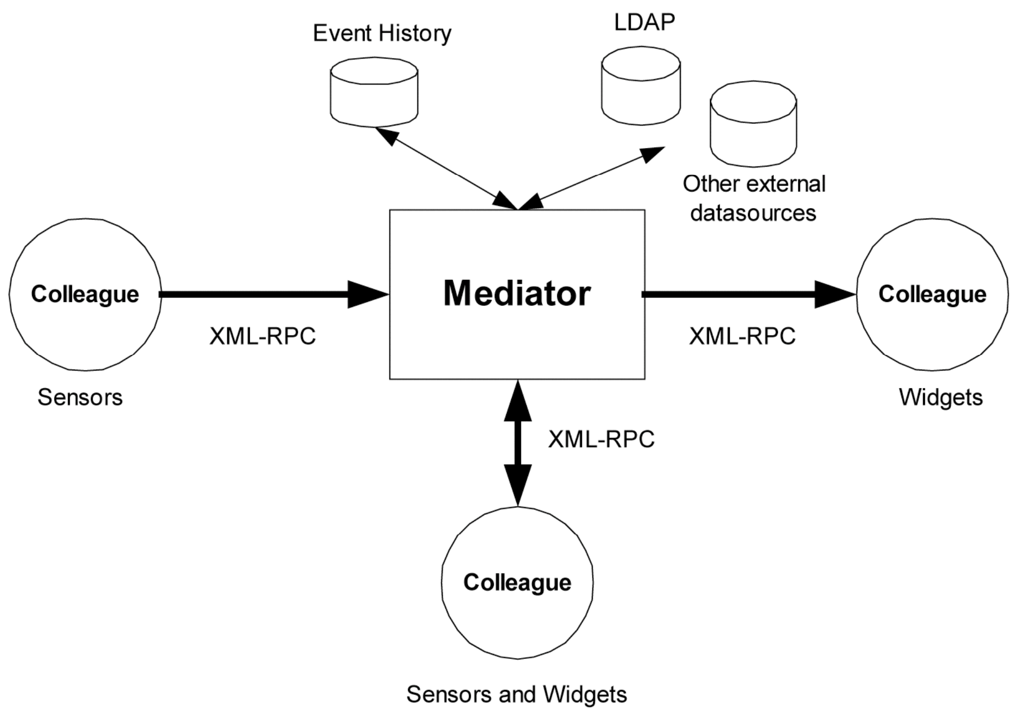

Figure 7. An instantiation of the CORBA mediator pattern.

As shown in Figure 7, our example the software architecture consists of a centralised mediator facilitating the communication among the colleagues. In this example we have three types of colleagues (according to the description above): a colleague solely consisting of one or more sensors, a colleague solely consisting of one or more widgets, and a colleagues consisting of both (i.e., which is able to both gather and present information). The communication in this implementation is done via XML-RPC—that is, the single components communicate via remote procedure calls (RPCs), and the representation of the data is done in the extended markup language (XML). All data represent events that happen in this architecture. We have different types. Examples are person-related events such as conversation messages, or presence information (e.g., status changes such as opening or closing a facet); and environment-related events such as changes in the environment (e.g., changes of prices at the stock exchange).

6 Related Work

Event and notification infrastructures such as Elvin [Segall & Arnold 1997], Khronika [Loevstrand 1991], and NSTP [Patterson *et al.* 1996] fulfil several of the above mentioned requirements. Elvin is a publish-subscribe notification service where consumers use content-based addressing to select notifications of interest. The developers consider the application domain for Elvin in area of network management, legacy application integration and as an infrastructure for computer-supported cooperative work. Fitzpatrick *et al.* [2002, p.448] point out that “Elvin supports awareness because it allows us to augment the workaday world and give imperceptible computer-based events a form that can be made publicly available and accessible as an informational resource for perception.”

Khronika is an event browsing and notification system that addresses the problems of information overload and information distribution. Loevstrand *et al.* built the Khronika shared event-notification system in order to "increase peoples' awareness of what is going on around them over time by improving the effectiveness in which event information is dispersed in a work community." (p.266). Key elements are events, daemons, and notifications. Initial motivation for Khronika was the information overload caused by the massive distribution of undirected event-information, such as through email list. The system provides a central service for the management of such events. Users can express their interest in events by constraints that are observed by daemons. Instead of being swamped with email announcement users can select the information they want to be aware of.

So, in Elvin and Khronika the primary focus is on the technical distribution of information—that is, these infrastructures deal with incoming and outgoing information (requirement 1), in an efficient way (requirement 2). Through the locales framework, which is applied in Elvin and Orbit, the situational context (part of requirement 3) of the users are taken into consideration for the selection of the information [Fitzpatrick *et al.* 2002]. However, the remaining requirements are not completely addressed—particularly the self-awareness (requirement 4). Prinz *et al.* [2002] point out that "it is necessary to develop an open infrastructure that supports the exchange of activity information and awareness across the boundary of different applications…", which should be able to "visualise activity information using different presentations that can as well be integrated into different applications" (p.91). Elvin and Khronika support this application-independent approach.

In the are of context-aware applications, some concepts and implementations of context support have been presented in the area of Human-Computer Interaction, and Ubiquitous Computing. Schilit *et al.* [1994] point out that for context models the following questions have to be answered: where you are, who you are, and what resources are nearby. Later, Dey [2000] developed the Context Toolkit and took the following parameters for modelling contexts: location of the user, identity of the user, time, and environment or activity. Although these models are very elaborated and offer good solutions for the support of situational and interpersonal contexts (requirement 3), they primarily address the single-user and do not support cooperative settings and shared contexts.

Some examples of context-like models for cooperative settings are the following: The AREA system describes situations as relationships among objects, where objects are single persons, artefacts, or aggregations such as groups of people. Users can specify which events and artefacts they are interested in and when and in which intensity they want to be informed [Fuchs 1999; Sohlenkamp *et al.* 2000]. The Atmosphere model [Rittenbruch 1999] describes contexts as 'spheres'. Users classify their actions on artefacts by means of 'contextors' and map them to specific contexts. When an action is performed a pre-defined contextor has to be selected. The AETHER model defines the relations between objects using semantical networks [Sandor *et al.* 1997]. The Model of Modulated Awareness (MoMA) applies a reaction-diffusion metaphor. This metaphor is based on the idea that whenever two or more entities have contact their state is modified in some way. Group awareness is produced and consumed through fields [Simone & Bandini 2002].

These latter models provide good support for cooperative situations. They allow a precise model with close correspondence to the modelled part of the reality, as well as a clear mapping of real events and situations to parts of the model. However, the modelling effort is quite high and typically these models do not adapt to changes in the modelled part of the reality. The AREA-model is quite precise and entails no additional effort for its users, but fails sometimes to map events and situations of the reality to the model. It is difficult to adapt the model to changes in reality. The Atmosphere model supports detailed modelling and succeeds in mapping events and situations to the model. However, the adaptation of the model is very difficult and the system requires considerable additional effort by the user. For AETHER and MoMA the situation is the same: they are very elaborated and provide the most precise modelling of reality, and they entail no additional effort from the users. The mapping of events and situations to the model is in general more adequate than in AREA, but less adequate than in Atmosphere. However, because of the complex models and mechanisms, the adaptation to changes in the reality is extremely high.

In the TOWER environment contexts are implemented as an extension of the event and notification infrastructure (ENI) [Gross & Prinz to appear]. Events are the basis for the processing of the awareness information and the context information in TOWER. The events are produced by sensors, which are associated with actors, shared material, or any other artefact in the electronic and physical environment that might be interesting for the individual user or the social context. Events are described as strings of attribute-value pairs. For instance, `producer=bill.gates&artefact=longhorn`. The sensors send the events they capture to the ENI server. The ENI server stores and administrates the events. The context module and the situation module of the ENI server are responsible for the context processing. The context module analyses the attributes of incoming events and compares these attributes with the context descriptions in the context database. If all or some attributes match, the context module attaches a context attribute to the incoming event (e.g., `event-context=ProjectX`). On the other side the situation module analyses the attributes of the events a user produces through her specific behaviour and tries to reason about the current work context of the respective user. The system can then compare the user's current work context with the incoming events' context of origin and provide the user with information that is important in her current situation—that is, the ENI server sends the respective events to the users' indicators.

The context model of TOWER and ENI supports incoming and outgoing information (requirement 1), in an efficient way (requirement 2). It takes situational parameters and interpersonal parameters into consideration (requirement 3) and supports self-awareness (4). It supports primarily synchronous modes, but asynchronous support is also possible (requirement 5). Concurrent contexts are supported (requirements 6). Compared to the approach presented in this paper, the main weakness lies in privacy protection: although, events have an access control mechanism, there are no such fine-grained mechanisms such as constraints or facets.

Finally, denoting the consideration of constraints in their work on social translucence, Erickson and Kellogg [2000] suggest constraints affecting the fields of vision, hearing, and speech. Also Chat Circles [Donath & Viegas 1999] relies on physical constraints for its spatial localisation of conversations implementing a hearing range. Yet, physical constraints are not the only category.

7 Conclusions

"The increasing importance of designing spaces for human communication and interaction will lead to expansion in those aspects of computing that are focused on people, rather than machinery" [Winograd 1997]. With this paper we provided a simple but powerful mechanisms for people to control their awareness in the digital realm—based on real world mechanisms. It helps us to get away from accidentally managing awareness and underlines the importance of social aspects and concepts based on social reality.

Nowadays, e-mail is the predominant CMC application, but believing recent surveys, this is going to change quite soon. A Giga Information Group planning assumption states: "Real-time communication [in this case instant messaging] with awareness will become a mission-critical business tool by the end of 2004, with instant messages surpassing e-mail traffic by 2006" [Rasmus 2002]. With the spreading of "awareness aware applications" awareness management is going to become an extremely important issue. We take a first step in this direction realising basic design challenges. Most of them can be addressed in some way, yet one remains a major problem: the effort. Due to today's human-computer interfaces its is hardly possible to achieve a nearly effortless and implicit awareness management in the digital realm.

References

Antunes, M., Silva, A.R. and Martins, J. An Abstraction for Awareness Management in Collaborative Virtual Environments. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST 2001 (Nov. 15-17, Banff, Canada). ACM, N.Y., 2001. pp. 33-39.

Beaudouin-Lafon, M. and Karsenty, A. Transparency and Awareness in a Real-time Groupware System. In Proceedings of the ACM Symposium on User Interface Software and Technology - UIST'92 (Nov. 15-18, Monterey, CA). ACM, N.Y., 1992. pp. 171-180.

Benford, S.D. and Fahlen, L.E. A Spatial Model of Interaction in Large Virtual Environments. In Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW'93 (Sept. 13-17, Milan, Italy). Kluwer Academic Publishers, Dortrecht, NL, 1993. pp. 109-124.

Bogdan, C. and Sundblad, O. A Cue-Based, Integrated System for Supporting Social Awareness. http://cid.nada.kth.se/pdf/cid_45.pdf, 1999. (Accessed 12/11/2004).

Boyd, D. Faceted Id/Entity: Managing Representation in a Digital World. Master's thesis, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA, Aug. 2002.

Buergy, C. and Garrett, J. Situation Aware Interface Design: An Interaction Constraints Model for Finding the Right Interaction for Mobile and Wearable Computer Systems. In International Symposium on Automation and Robotics in Construction - ISARC 2002 (Sept. 23-25, Washington, DC). 2002. pp. 536-568.

Dey, A.K. Providing Architectural Support for Building Context-Aware Applications. Ph.D. thesis, Georgia Institute of Technology, Nov. 2000.

Donath, J. and Viegas, F. Chat Circles. In Proceedings of the Conference on Human Factors in Computing Systems - CHI'99 (May 15-20, Philadelphia, PE). ACM, 1999. pp. 9-16.

Dourish, P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. In Proceedings of the Conference on Computer-Supported Cooperative Work - CSCW'92 (Oct. 31-Nov. 4, Toronto, Canada). ACM, N.Y., 1992. pp. 107-114.

Endsley, M.R. Design and Evaluation for Situation Awareness Enhancement. In Proceedings of the Human Factors Society -- 32nd Annual Meeting (Oct. 24-28, Santa Monica, CA). 1988. pp. 97-101.

Erickson, T. Short Talk: Designing Visualisations of Social Activity: Six Claims. In Extended Abstracts of the Conference on Human Factors in Computing Systems - CHI 2003 (Apr. 5-10, Fort Lauderdale, Florida). ACM, N.Y., 2003. pp. 846-847.

Erickson, T. and Kellogg, W.A. Social Translucence: An Approach to Designing Systems that Support Social Processes. ACM Transactions on Office Information Systems 7, 1 (Mar. 2000). pp. 59-83.

Fitzpatrick, G., Kaplan, S., Mansfield, T., Arnold, D. and Segall, B. Supporting Public Availability and Accessibility with Elvin: Experiences and Reflections. Computer Supported Cooperative Work: The Journal of Collaborative Computing 11, 3-4 (2002). pp. 447-474.

Fuchs, L. Situationsorientierte Unterstuetzung von Gruppenwahrnehmung in CSCW-Systemen. Ph.D. thesis, Institute for Applied Information Technology, GMD--German National Research Center for Information Technology, St. Augustin, Germany, 1998.

Fuchs, L. AREA: A Cross-Application Notification Service for Groupware. In Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work - ECSCW'99 (Sept. 12-16, Copenhagen, Denmark). Kluwer Academic Publishers, Dortrecht, NL, 1999. pp. 61-80.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1994.

Goffman, E. The Presentation of Self in Everyday Life. Anchor, N.Y., 1956.

Gross, T. and Prinz, W. Awareness in Context: A Light-Weight Approach. In Proceedings of the Eights European Conference on Computer-Supported Cooperative Work - ECSCW 2003 (Sept. 14-18, Helsinki, Finland). Kluwer Academic Publishers, Dortrecht, NL, 2003. pp. 295-314.

Gross, T. and Prinz, W. Modelling Shared Contexts in Cooperative Environments: Concept, Implementation, and Evaluation. Computer Supported Cooperative Work: The Journal of Collaborative Computing (to appear).

Gutwin, C. and Greenberg, S. A Descriptive Framwork of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work: The Journal of Collaborative Computing 11, 3-4 (2002). pp. 411-446.

Hudson, S.E. and Smith, I. Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems. In Proceedings of the ACM 1996 Conference on Computer-Supported Cooperative Work - CSCW'96 (Nov. 16-20, Boston, MA). ACM, N.Y., 1996. pp. 248-257.

Loevstrand, L. Being Selectively Aware with the Khronika System. In Proceedings of the Second European Conference on Computer-Supported Cooperative Work - ECSCW'91 (Sept. 24-27, Amsterdam, NL). Kluwer Academic Publishers, Dortrecht, NL, 1991. pp. 265-278.

Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D. Intelligent Information-Sharing Systems. Communications of the ACM 30, 5 (May 1987). pp. 390-402.

Merriam-Webster, I. Merriam-Webster Online. http://www.m-w.com/, 2004. (Accessed 4/6/2004).

Patterson, J.F., Day, M. and Kucan, J. Notification Servers for Synchronous Groupware. In Proceedings of the ACM 1996 Conference on Computer-Supported Cooperative Work - CSCW'96 (Nov. 16-20, Boston, MA). ACM, N.Y., 1996. pp. 122-129.

Preece, J., Rogers, Y. and Sharp, H. Interaction Design: Beyond Human-Computer Interaction. Wiley, N.Y., 2002.

Prinz, W., Graether, W., Gross, T., Kolvenbach, S., Klein, K.-H., Pankoke-Babatz, U. and Schaefer, L. TOWER: Presenting Activity Information in a Theatre of Work. In Supplement Proceedings of the ACM 2002 Conference on Computer-Supported Cooperative Work - CSCW 2002 (Nov. 16-20, New Orleans, LA). ACM, N.Y., 2002. pp. 91-94.

Rasmus, D. Evolving Toward Enterprise Instant Messaging. Planning Assumption RPA-052002-00008. Giga Information Group Inc., http://www.info-edge.com/toc_new/GA-6081toc.pdf, 2002. (Accessed 12/11/2004).

Rittenbruch, M. Atmosphere: Towards Context-Selective Awareness Mechanisms. In Human-Computer Interaction: Communication, Cooperation and Application Design - HCI'1999 (Aug. 22-26, 1999, Munich, Germany). Lawrence Erlbaum, Hillsdale, NJ, 1999.

Sandor, O., Bogdan, C. and Bowers, J. Aether: An Awareness Engine for CSCW. In Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work - ECSCW'97 (Sept. 7-11, Lancaster, UK). Kluwer Academic Publishers, Dortrecht, NL, 1997. pp. 221-236.

Sasse, M.A. Eliciting and Describing Users’ Models of Computer Systems. Ph.D. thesis, Faculty of Computer Science, University of Birmingham, Apr. 1997.

Schilit, B.N., Adams, N.I. and Want, R. Context-Aware Computing Applications. In Proceedings of the Workshop on Mobile Computing Systems and Applications (IEEE Computer Society, Santa Cruz, CA, 1994. pp. 85-90.

Schmidt, K. The Problem With Awareness: Introductory Remarks on Awareness in CSCW. Computer Supported Cooperative Work: The Journal of Collaborative Computing 11, 3-4 (2002). pp. iii-iv.

Segall, B. and Arnold, D. Elvin has Left the Building: A Publish/Subscribe Notification Service with Quenching. In Proceedings of Australian UNIX and Open Systems Users Group Conference - AUUG'97 (Sept. 3-5, Brisbane, Australia). 1997.

Simone, C. and Bandini, S. Integrating Awareness in Cooperative Applications through the Reaction-Diffusion Metapher. Computer Supported Cooperative Work: The Journal of Collaborative Computing 11, 3-4 (2002). pp. 495-530.

Sohlenkamp, M. Supporting Group Awareness in Multi-User Environments through Perceptualisation. Ph.D. thesis, Institute for Applied Information Technology, GMD--German National Research Center for Information Technology, St. Augustin, Germany, 1999.

Sohlenkamp, M., Prinz, W. and Fuchs, L. PoliAwac - Design and Evaluation of an Awareness Enhanced Groupware Client. AI and Society - Special Issue on CSCW 14, 1 (2000). pp. 31-47.

Viegas, F. Collections: Adapting the Display of Personal Objects for Different Audiences. Master's thesis, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA, Aug. 1997.

Winograd, T. From Computing Machinery to Interaction Design. In Denning, P.J. and Metcalfe, R.M., eds. Beyond Calculation: The Next Fifty Years of Computing. ACM, N.Y., 1997. pp. 149-162.