Workspace Awareness for Distributed Teams

Johann Schlichter, Michael Koch, and Martin Bürger

Technische Universität München, Institut für Informatik,
D-80290 München, Germany

Abstract. Research in distributed problem solving in the last years focused on distributed applications which cooperate to accomplish a task. Another level of distributed problem solving is that of human teams which are distributed in space and cooperate in solving a problem. In this paper we will introduce distributed problem solving from the 'human level', briefly present the accompanying research area of Computer-Supported Cooperative Work (CSCW) and the different basic mechanisms of computer support for workgroup computing, and then focus on the awareness information that is of special importance for supporting coordinated cooperation of groups with unstructured tasks.

1 Introduction

The emergence of high-speed local area computer networks at the beginning of the 1970s resulted in distributed systems becoming an important topic in computer science. As a sub-discipline of distributed systems, distributed problem solving (DPS) emerged by combining the ideas of distributed systems and those of artificial intelligence.

Distributed problem solving can be defined as the cooperative activity of several decentralized and loosely coupled problem-solvers that act in separated environments. Hence, one generally assumes a number of instances, the problem-solvers, that are distributed in space and collaborate in completing a common task.

Most approaches assume that the cooperating instances are software components, programs or agents. Important issues at this 'application level' are group communication, RPC, concurrency control, replication, and distributed objects (see [8,29] for more details).

The increasing network availability in the last decade has not only been an enabling factor for building distributed systems with cooperating applications, it also has been a major breakthrough enabling distributed group work using computer and network technologies. Hence, the scenario of distributed applications trying to solve a given task is only one possible viewpoint of distributed problem solving. Another very important viewpoint is to investigate distributed human teams that collaborate to achieve a common goal using computers connected through a communication network.

This viewpoint is very important because nowadays the members of human teams are often spread among several departments or companies. Computer technology and increased network availability has enabled and improved distributed group work. The coordination of the contributions of the team members is an important task in supporting distributed group work.

The collaborating teams often use distributed applications for their work. Coordination tasks in particular are carried out through the distributed software system (application level). Figure 1 shows this relationship of human level cooperation and application level cooperation. In this paper we will focus on the human level.

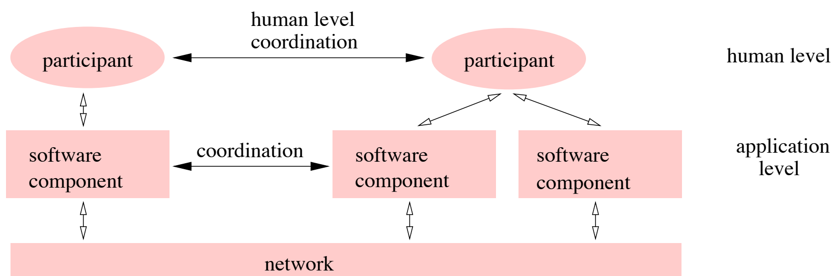

Fig. 1. Different levels of distributed problem solving in teams.

The rest of this paper is organized as follows: In Section 2 we will give a brief overview of the basics in collaboration among people and in the research area that focuses on providing computer support for collaboration. Then we will summarize the basic support mechanisms from the human level view (Section 3). Thereby, we will also touch upon the application level details that are needed for providing support at the human level. In Section 4 we will focus on one basic mechanism for supporting coordination in collaborative work on unstructured tasks: group awareness. The basic concepts listed in Section 4 will be highlighted in the context of providing support for distributed teams collaborating on writing documents in Section 5. In this section we will present the mechanisms the group editor environment IRIS provides for supporting workspace awareness as an example.

2 Collaboration and Computer Support for Collaborative Work

2.1 Collaboration at the 'Human Level'

The terms 'cooperation' or 'collaboration' are used to refer to a set of participants working together to produce a product or service [3, p.362]. Collaboration requires two or more participants who contribute to a common task. A crucial point for successful collaboration is the manner in which individual work is related to the group as a whole. Co-workers make autonomous decisions when working alone, under changing and unpredictable conditions, which the group cannot foresee or plan for. To enable a separated group of co-workers to collaborate, they need to coordinate themselves [27]. The importance of coordination can be seen in the need to bring the efforts of all co-workers together in order to produce a product or service. Examples of the need for coordination in collaborative work is the need to ensure the completion of all work, the lack of redundant work (e.g. avoid conflicting actions) and the timely completion of the work.

Communication mechanisms are critical to coordination and thus essentially needed for collaborative work. To perform communication participants typically use two fundamental human skills [4]:

- direct communication with other participants and
- manipulation of shared artifacts.

Typically, manipulation of shared artifacts can be observed by other participants, thus constituting a form of indirect communication. These skills are often used in combination. For example, when communicating directly participants often use references to shared artifacts as an easy way of establishing referential identity [7]. Similarly, when working with shared artifacts, participants often communicate directly with each other.

Figure 2 shows the different communication and coordination channels among the participants considering that in most cases the participants are working on shared artifacts and some of the communication and coordination is done via manipulation of the shared artifacts.

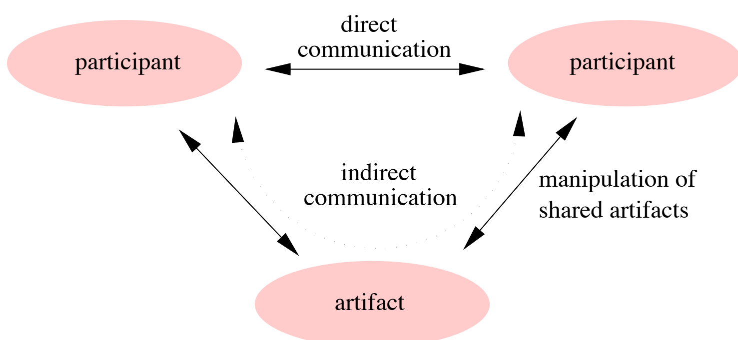

Fig. 2. Cooperation with shared artifact (adapted from [28]).

2.2 CSCW and Groupware

Computers and the emerging networks can be used to support collaboration. The research area that is concerned with computer support for collaborating teams is called "Computer-Supported Cooperative Work" (CSCW). The term

CSCW was coined by Irene Greif (Massachusetts Inst. of Technology) and Paul Cashman (Digital Equipment Corporation) to explain the scope of a small workshop with attendants from different fields [2,19]. CSCW is not a self-contained research area with its own technology but an interdisciplinary area of study within which the main issues are to understand collaboration and to integrate different technologies in order to support collaborating teams. Wilson defines CSCW in [36] in the following way:

> *"CSCW is a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking and associated hardware, software and techniques."*

While CSCW is the name for the research area, the term *groupware* stands for the systems that support group work. In practice this means that groupware is software and/or hardware which implement the theoretical foundation of CSCW activities. Johansen writes [20]:

> *"Groupware is a generic term for specialized computer aids that are designed for the user of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support."*

A major issue in CSCW is to understand the way in which computer systems can be instrumental in reducing the complexity of coordinating cooperative activities. Groupware reflects a change in emphasis from using the computer to solve problems to that of facilitating human interaction [13].

Groupware can be designed to support a face-to-face group or a group that is distributed over many locations. Furthermore, a groupware system can be built to enhance collaboration within a real-time interaction, or an asynchronous, non real-time interaction. These time and space considerations suggest the separation of the groupware domain into four quadrants based on whether users are working at the same place or different places and whether they are working synchronously or asynchronously, as shown in Figure 3[1].

As listed in the examples in Figure 3 groupware provides support for many functional areas. There are now numerous examples of both commercial products and research prototypes for most of the major categories of CSCW technologies (the unOfficial Yellow Pages of CSCW lists 340 commercial or experimental groupware systems [26]). For example there are media spaces for real-time communication, email systems for asynchronous communication, groupware tools for cooperative development of documents, drawings, software for meeting facilitation, and tools for workflow management (see [6,9,13,26] etc. for references to concrete systems).

[1] The figure is called 'Any-Time Any-Place' matrix because it shows that groupware may bridge space and time constraints and enable collaboration at any time from any place.

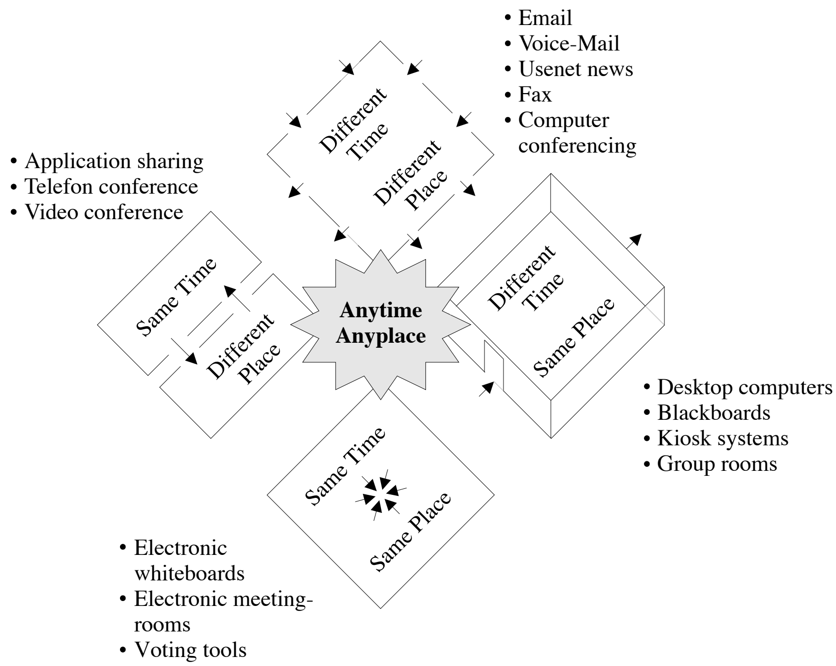

Fig. 3. Any-Time Any-Place Matrix (adapted from [21]).

Numerous groupware products are now in use in many commercial environments, for example, Lotus Notes, Microsoft Exchange, IBM's WorkGroup, Novell's GroupWise, Collabra Share, etc. These systems typically integrate a number of tools for communication, workflow, database-sharing, contact management, and group scheduling, and operate across a variety of environments.

3 Support Mechanisms for Collaborative Work

The cooperation of people who engage in a common task requires the coordination of the task-related activities as well as the coordination of the resources used during the execution of these activities. As already mentioned above the coordination at the human level is often implemented by a distributed software system. The components of this system negotiate with each other at the application level to achieve the desired coordination behavior at the human level. In this section we first discuss the main mechanism to support coordination from the human perspective and second, how this mechanism may be implemented at the application level. Thus, from the mechanism required at the human level we derive the desired application level support.

Essential for successful collaborative work is the efficient communication between co-workers. This is especially important if the group work is distributed across space and time. Communication serves two main needs (see Figure 4):

- on the data level to exchange shared information, e.g. the exchange of group documents or group membership information, and
- on the relationship level to coordinate the group activities as well as the access to and the usage of shared resources.

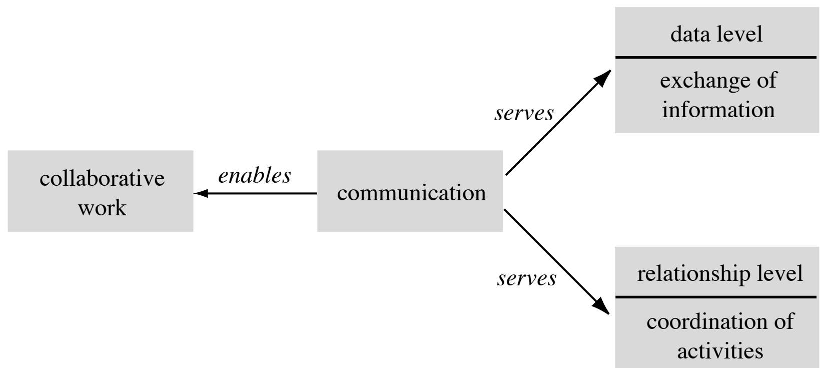

Fig. 4. Support of collaborative work on the human level.

The exchange of information may be achieved by direct or indirect communication (see Figure 2 in Section 2). In the first case a direct communication link between the involved group members is established; the information is exchanged along this link either asynchronously (e.g. email) or synchronously (e.g. video conferencing). Indirect communication assumes a shared information space which may be used to exchange and propagate information. Information stored in the shared information space may be accessed and retrieved by other group members. A typical example of this communication type is a bulletin board system where the communication link between the cooperating partners is achieved via the shared information space. Indirect communication is only suitable for situations where co-workers cooperate loosely.

For both direct and indirect communication the coordination dependencies among co-workers are not explicitly defined. The coordination task is fuzzy and it can only be supported by exchanging sufficient information to get a mutual understanding of the progress and the current situation of the group work.

The second aspect of communication at the human level refers to explicit coordination. In this case, the dependencies among co-workers are well-defined and explicitly specified. Communication in this context intends to initiate transitions within group work states. Examples are the notification of co-workers after documents have been modified or the hand-over of a circulation folder to another co-worker. Notifications might inform another co-worker that a group document has reached a certain state in which he can start working on it.

These two communication aspects at the human level can be supported by the following technical mechanisms at the application level (see Figure 5) :

1. direct communication,
2. indirect communication.

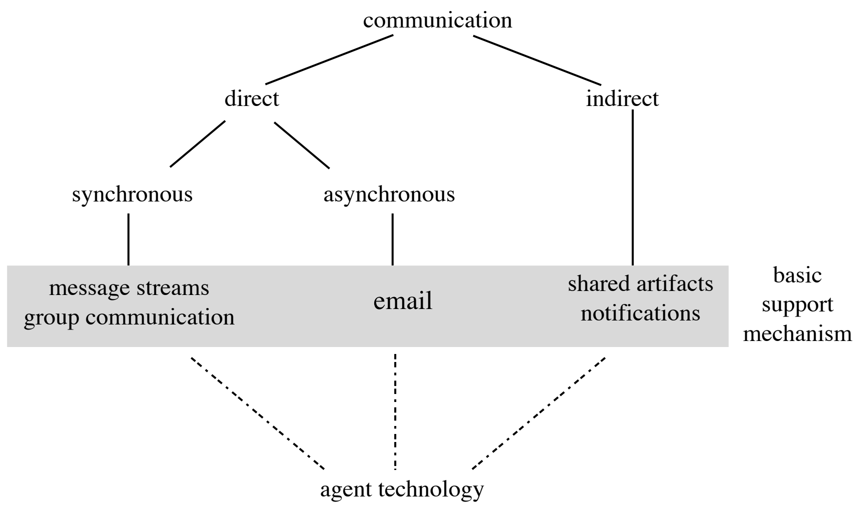

Fig. 5. Support of collaborative work on the application level.

3.1 Direct Communication

From the technical viewpoint direct communication handles the propagation and management of message streams between the involved co-workers. Services include the creation, transfer, synchronization and filtering of message streams. The direct communication can occur either synchronously or asynchronously. In the synchronous mode a real-time communication link is established and the information is exchanged in real-time between the involved co-workers. The video conferencing scenario in particular requires the synchronization of multiple message streams, such as audio, video and data. The data component itself may consist of multiple streams; for example one data stream contains the shared group document while other data streams may specify telepointer coordinates of the different group members. The communication has to be adapted to the available network bandwidth and to the quality of service requirements as well as to the characteristics of the coordination tasks. Examples of the latter case are the urgency of the coordination or the relationship between the group members, e.g. does there already exist a trusted relationship between them because they cooperated successfully in the past? A trusted relationship requires a much smaller communication channel because there is already a mutual understanding between the involved partners. Thus, it is not necessary to specify explicitly all coordination goals and requirements.

Asynchronous communication is typically based on email, i.e. there is no real-time interaction between the involved co-workers. Thus, the requirements for the necessary communication infrastructure are less stringent than for the synchronous communication.

An increasingly important technical concept to model and implement direct communication is that of group communication [8,35]. Every co-worker is represented by a separate system component which is executed in the local environment. All these components are combined into the single abstraction of a group. Messages sent to the group are automatically distributed to all group components and thus, delivered to all co-workers. Besides message delivery, group communication incorporates further mechanisms, such as group management, ordering of messages and atomic message delivery despite network or host failures.

In the context of direct communication, information is actively propagated to the co-workers. Coordination of group activities is based on events caused by the information flow.

3.2 Indirect Communication

Indirect communication is based on the existence of a central or a distributed workspace which contains all the shared artifacts of the group work. The work space is passive and it may be browsed or queried to determine group documents for retrieval. Modifications of shared artifacts may cause notifications which are propagated to interested co-workers.

Shared Artifacts Inherent with group work is the existence of a shared context which consists of a shared environment as well as a multiuser interface to this environment. The environment incorporates a variety of shared artifacts which are managed and manipulated by the whole group. A simple example of a shared artifact is the group document which is jointly authored by a group of authors. Besides different views of the same object, shared artifacts must also allow a wide variety of working modes ranging from individual to collective work. A CSCW system must support a seamless transition between these working modes as well as between different activities operating on these artifacts.

The application level provides functionality to manage and to handle the access to the shared artifacts. Two important functions are concurrency control and replication management. Both functions must be supported transparently at the human level. Concurrency control handles the concurrent access of multiple users to shared information in order to preserve information consistency. Concurrency control mechanisms may be categorized into pessimistic and optimistic approaches.

Pessimistic approaches attempt to keep the documents consistent at all situations even in the case of network partitionings. Well known examples of these approaches are access locks and transactions. Past CSCW-systems favored the pessimistic approach, but recently optimistic approaches gained more interest in the CSCW community, especially when mobile workers are included in the shared environment. In the latter case, it is not possible to assume a permanent network connection between all involved co-workers. Optimistic approaches assume that the cooperating co-workers are less likely to get into conflict while manipulating the shared information. Additionally, people use social protocols to avoid any conflicting actions, e.g. a person announces via audio link to his co-workers that he will manipulate a certain paragraph. Pessimistic approaches apply technical protocols in order to avoid conflicts while the optimistic protocols prefer social protocols to ensure information consistency. Social protocols are less restrictive and offer group members more freedom to adjust the concurrency control to the characteristics of the group and its current state. Social protocols need a high degree of group awareness in order to be effective.

Closely related with concurrency control and the distributed environment is the replication of information. Replication improves the access times as well as the availability despite network or machine failures. Besides the internal consistency of a single copy this requires additionally the mutual consistency of the copies. The concurrency control mechanism must be extended to handle multiple copy consistency.

Notifications Notifications are an essential means of improving group awareness. They are based on messages which are generated automatically through user interaction with the workspace. For example, the modification of a paragraph by one co-worker might generate a notification which is then sent via multicast to all other co-workers. The notification mechanism and its relationship to the coupling mode of the co-workers will be discussed in more detail in later sections.

At this point we can summarize by stating that communication at the human level which is required in order to achieve group coordination is implemented at the application level by direct and indirect communication.

3.3 Agent Technology

For the design and implementation of the application level mechanisms we can apply agent technology. Thus, the distributed system consists of a number of agents which operate autonomously and cooperate with each other to perform the global task. The knowledge and strategies needed to solve the global task are distributed across the individual agents. Basic agent functions are the execution of assigned subtasks and the communication with other agents. The latter function corresponds to the communication at the human level.

Besides the well-known general usage of agents in distributed systems, agents may be applied to handle some specific coordination tasks in the context of CSCW. One important usage is the filtering of messages [24]. This can be done at the sender's side according to his privacy needs, at the receiver's side according to his interests, or at the organizational level according to basic company policies.

The second potential usage of the agent technology refers to the implementation of concurrency policies depending on the desired quality of service and the currently available network infrastructure. Agents continuously monitor the current network traffic and the type of interaction between co-workers. For example, in the case of network partitionings agents switch from pessimistic to optimistic concurrency models to enable work progress despite interrupted message traffic. The policies may also change according to the coupling modes of the co-workers. In the case of loose coupling the system might prefer technical protocols while in situations of tight coupling the system might switch to social protocols.

Another important aspect where agents may be applied is the delegation of tasks. Agents interact and negotiate with each other to determine a suitable contracting agent. The contract net model [34] provides a suitable general protocol to design and implement this negotiation process. If agents are representatives of users then the negotiation process at the agent level may result in the delegation of activities at the human level.

4 Awareness for supporting collaboration

4.1 Coordinating Unstructured Work

There are two different extreme types of tasks which should be considered when talking of computer support for distributed teams (see Figure 6). On the one hand there are tasks which are done according to a standard procedure, such as that for approving business trips or processing bank credit applications. For these tasks one often can provide a detailed model that clearly describes the steps that are necessary to complete the task. On the other hand, there are tasks that are never done in the same fashion because they are inherently chaotic. Examples are creative work such as writing a paper for a conference or work that is subject to external influences, such as brokering shares. In contrast to the first task type there is no obvious structure. Single steps inside the task can only be described in a very high-level manner. In this paper we will address tasks that show a fine granular structure as *structured tasks* and tasks that do not show such a structure as *unstructured tasks*.[2]

Because these task types are so different, different mechanisms are needed to effectively support their coordination. A suitable mechanism for structured work is *explicit coordination*. The term 'explicit coordination' is chosen because the coordination is handled by actions explicitly initiated for coordinating the task. The initiator can be a software component that reacts upon a given task model by initiating some coordination actions. In the real world, this leads to the class of workflow systems, where the standard procedure is described by a model of the task.

For unstructured tasks, however, there is no abstract model of work that describes the steps that are necessary to complete a task. Instead, the system

[2] Practical tasks usually incorporate aspects of both, the structured and unstructured task models, and they are classified according to the dominating part.

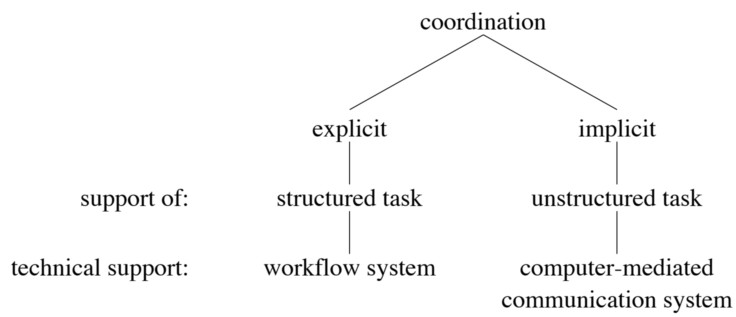

Fig. 6. Coordination for different types of tasks.

must offer as much freedom as possible to the co-workers so that they can do whatever they think is necessary to reach a particular goal. In this scenario, no automatic coordination is possible, because there is no predefined flow of work but coordination has to be adapted dynamically to the actual situation. This can not be done by an automatic system, but it must be done by the people themselves. This requires a high degree of group awareness where co-workers are aware of each other's past, current and possibly future activities within the shared environment. The awareness information needed can be exchanged by direct communication or by indirect communication (see Section 3) with the help of the system (computer-mediated communication). This whole process is called *implicit coordination*.

To summarize, one can say that awareness "*is part of the "glue" that allows groups to be more effective than individuals*" [18]. In the next subsection we will present an overview of the research results on awareness and their importance for coordination in group work.

4.2 Awareness Basics

The emphasis of much of recent research within CSCW has been to provide awareness-oriented collaboration systems where users coordinate their work based on the knowledge of what the members of the collaborating group are doing or have done. *Group awareness* can be defined as "*an understanding of the activities of others, which provides a context for your own activity*" [11].

An increase of awareness within a collaborating group has several advantages:

- It encourages informal spontaneous communication (e.g. via video conferences, phone calls, etc.), since people are more likely to use direct contact to others when they know their partner is not too busy and can be interrupted without interfering too much with the ongoing work (see for example PORTHOLES [11] and PEEPHOLES [17]).
- Awareness is important to keep group members up-to-date with important events and therefore contributes to their ability to make conscious decisions.

To determine how to support awareness by groupware it is helpful to distinguish between different sub-types of awareness. According to Greenberg [18], there are several types of group awareness needed to collaborate effectively:

- "**Informal awareness** *of a work community is basic knowledge about who is around in general (but perhaps out of site), who is physically in a room with you, and where people are located relative to you.*"
- "**Group-structural awareness** *involves knowledge about such things as people's roles and responsibilities, their positions on an issue, their status, and group processes.*"
- "**Social awareness** *is the information that a person maintains about others in a social or conversational context: things like whether another person is paying attention, their emotional state, or their level of interest.*" Other information can be the special skills a co-worker has.
- **Workspace awareness** is "*the up-to-the minute knowledge a person requires about another group member's interaction with a shared workspace if they are to collaborate effectively*".

It is rather simple for a groupware system to retrieve *a lot* of information about the interactions of the users with the workspace. But it is necessary to present this information in an adapted way to avoid swamping the users with useless information. So we will focus in the next subsection on models of workspace awareness that can be exploited in real systems to reduce the amount of information that is presented to the users.

4.3 Orientation Models for Workspace Awareness

There has been research on orientation models in collaborative processes. This is important to find new ways to present awareness information to the users (and to select which information is presented). One particular example is the GroupDesk project [1,15,16]. One result of this research was the discovery of the four modes of awareness [16]. These modes can be described by two orthogonal classifications:

- *Coupling*: There is the coupled awareness (participants have the same focus of work, e.g. they work on the same shared artifact and are aware of each other) and uncoupled awareness ("*information independent of the user's current focus of work*").
- *Synchrony*: Participants may either be aware synchronously (knowledge about events that happen currently) or asynchronously (knowledge about events in the past).

Together, these classifications result in the aforementioned modes of awareness which can be described by typical questions (see Figure 7).

When designing a real system, these modes must be treated differently. For example, uncoupled awareness information may be treated in a less obtrusive way at the user interface than coupled awareness information, which may be of

	synchronous	asynchronous
coupled	What is currently happening in the actual scope of the work?	What has changed in the actual scope of the work since last access?
uncoupled	Things of importance which occur currently anywhere else?	Has anything of interest happened recently somewhere else?

Fig. 7. Modes of awareness (adapted from [16]).

a greater interest to the collaborating person. Coupled awareness information may be presented by a pop-up message (which can be very intrusive), while uncoupled awareness information can be presented by changing the color of an icon or by printing a short message in a status bar. The same applies to synchronous vs. asynchronous awareness, where it would be desirable to have some sort of summary about past events, so that one can catch up quickly to the current state of things without wading through too many details.

Another more sophisticated awareness model was presented by Rodden [32,33]. This model describes the interactions of users with a shared workspace in terms of a spatial metaphor (like in a virtual reality meeting place). The most important terms here are *nimbus* and *focus*:

- *Nimbus* describes the location(s) that a user is occupying in the workspace.
- *Focus* describes the location at which the user is looking (may be several places at the same time).

The nimbus and focus of users in a workspace can then be used to describe how (and if) two users are aware of each other. The strength of awareness user A has of another user B can be described by the amount of overlap between A's nimbus/focus and B's nimbus/focus. Some simple examples in a two-dimensional space can be seen in Figure 8.

Is is easy to see that these different situations should result in a different strength of awareness of the two users of each other. These strengths could then be used in an implementation to select how (and if) actions of B should be presented to A and vice versa.

4.4 Filtering Awareness Information

There are some reasons why the groupware system should filter the awareness information before it is brought to the user's attention:

- Avoidance of information overflow by adapted presentation as described above by the orientation models.
- Privacy issues, which will be described next.

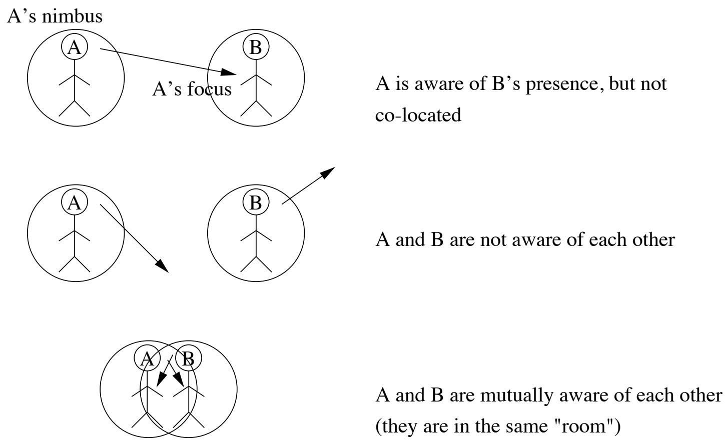

Fig. 8. Overlap of nimbus and focus.

There is one trap for an unwary designer of groupware systems with awareness mechanisms: the issue of privacy may arise (see [5,37] for more). This can lead to acceptance problems of the system, because people may not want the system to look over their shoulder and distribute everything it sees to other people. These problems are not likely to occur in a small group of socially equal persons (like in joint editing of a conference paper), but this can be a very serious issue for larger systems spanning several departments within a big company.

Mechanisms to reduce the probability of rejection by the users can be:

- Show users what others see of them.
- Give users control about information that is sent to others by means of an outgoing filter that filters all information that is broadcast to the other co-workers.

In the next part we will present group editors as an application area for group awareness and discuss the application of workspace awareness for supporting coordination in collaborative writing groups.

5 Awareness Information for Supporting Coordination in Collaborative Writing

5.1 Collaborative Writing

One of the most common type of tasks undertaken by groups is the collaborative editing of documents. Additionally, this task is probably the most suitable for CSCW applications because computer systems are already adept at document manipulation.

Collaborative writing is defined in [25] as a '*process in which authors* [e.g. editors, graphics experts, users, reviewers] *with differing expertise and responsibilities interact during the invention and revision of a common document*'. Regarding the types of cooperative tasks introduced at the beginning of Section 4 collaborative writing in general is an unstructured task. One cannot rely on a fine granular work plan that shows all interaction needed for coordination.

Different writing strategies can be identified in writing groups (see [31] for some examples). The strategy most commonly identified with collaborative writing is the 'separate writers strategy': Work begins with a division of work and responsibilities. Then the co-authors produce their parts of the document separately. This asynchronous phase can last a long time and is usually interrupted by several synchronous coordination meetings. Finally, the parts are distributed for annotating and for assembling into the final document. Sometimes the co-authors work very closely together (start-up meeting, regular coordination meetings, spontaneous conferences) but most of the time the authors work on their own.

Software applications supporting collaborative writing are called *group editors*. Many tools have already been proposed to support collaborative writing for different media. Examples for such group editors are QUILT [14], GROVE [12] and PREP [30].[3]

5.2 Awareness Information in Group Editors

As for all unstructured tasks, information and awareness are very important for successful collaboration in group writing. Therefore, support for achieving awareness has been an issue in many group editor projects[4]. A brief introduction in the usage of awareness information in several group editors can be found in [10].

In a group editor we have one major shared artifact, the common document with additional information such as annotations. The co-authors usually work asynchronously on parts of the document. The interaction of the authors with the common workspace can be used to construct different awareness information. One can distinguish events and status information:

- *Events* hold information on a particular action that has happened. Events are distributed to users and filtered on demand (see Section 4.4). Examples of events are notifications about a document change or about the login of a user.
- *Status attributes* are gained by combining events to form some kind of longer-lived information. A status might be a list of active users, a list of working areas or the reachability of certain hosts on the network.

[3] See [22] and [23] for more examples of commercial and academic group editors.

[4] Most of the other issues were not related to human level coordination but to application level coordination like concurrency control with replicated document data.

In addition to events and status attributes one has to mention the history (a log of changes to the document) as a source of awareness.

To investigate the notions of awareness, we have developed a group editor environment called Iris [22,23]. In the following subsection we will briefly present the method of implementation of support for group awareness in Iris.

5.3 Awareness Information in Iris

The core of the Iris system consists of several replicated components that communicate with each other to ensure document consistency and to calculate and distribute awareness information. This core service is called 'storage and awareness service'.

The service provides access to the document itself and to a history of document changes. Additionally, notification event are generated from the interaction of the user applications with the document data, and status attributes are distributed. A status attribute consists of a name and a value. The awareness service stores attributes for every document, for every user and for every host participating in the editing process.

Most of the status attributes are set automatically by the awareness service according to the user actions. The following listing describes the standard attributes for documents and users as defined in Iris. (For more information on the attributes and on the implementation of the storage and awareness service of Iris see [23].)

- *Document information*: Automatically generated standard attributes for documents are lists of read- and write-work areas[5], a list of the hosts that store replicas of the document and a list of the users who accessed the document. In addition to the main information all these lists store the time of the last change.
- *User information*: For every user that is or has been working with documents the system maintains a status attribute (possible values are 'active-in-groupware-application', 'idle-in-groupware-application', 'active-on-host', 'idle-on-host', 'inactive', 'no-info'). Other attributes calculated by the system are a list of hosts and of the documents the user has been working on (the last-changed time of the list elements provides information on when the user has last worked on the document/host).

In addition to the information calculated from the interaction of the users with the workspace, the user applications may set additional status attributes. This is important for information that is valuable to the group process but which cannot be determined from the interaction with the workspace. User-defined standard attributes are a document status attribute, and a list of reservations[6].

[5] The 'work areas' store information on document positions that different users have been working in within a configurable period of time.

[6] A 'reservation' can be seen as an optimistic lock. Reservations can be set for parts of a document and are displayed by the editing applications.

User defined user attributes are a name attribute, a list of roles the user has for different documents, and an additional status value (e.g. 'on leave', 'do not disturb', 'in meeting'). This additional status value supersedes the automatically calculated user status when defined. The user may also provide a validity time and a comment for the user defined status value.

Changes in the document information or in the status attributes are distributed as events. The events are automatically generated from the interaction of the users with the shared document data. At present there are no means for filtering events at the event source or at the event receivers.

5.4 Awareness Information at the User Interface

User interface applications use the storage service to access the document and to access the awareness information.

At present the user interface applications offer special support for displaying the status attributes only. Events are just listed in extra windows[7].

Figure 9 shows a screenshot of the structure editor, a tool for displaying the structure of a document. This tool displays the document structure and provides functionality for editing the structure, for requesting information on (sub-)documents and for launching editor applications for (sub-)documents.

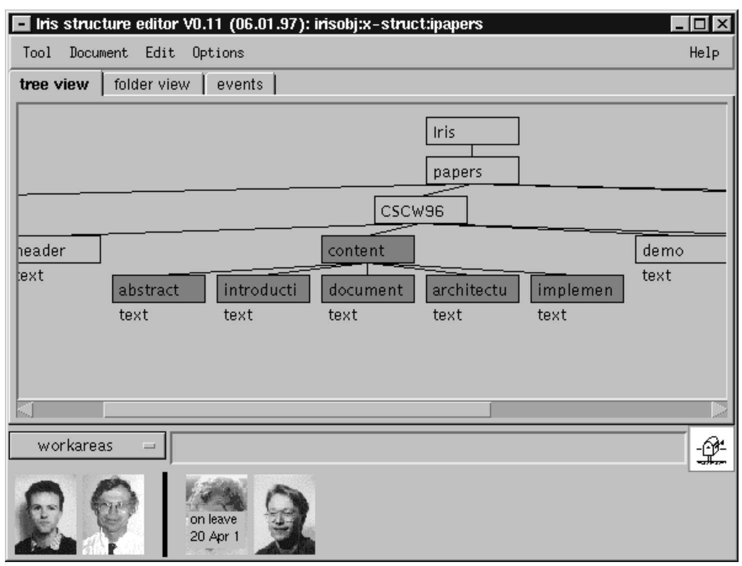

Fig. 9. IRIS document structure editor.

---

[7] A service for a configurable adapted display of events as mentioned in the previous section is in progress.

In the structure display the value of status attributes which are related to sub-structures of the document is visualized. It is possible to display different types of work areas and reservations in the tree display. The nodes are colored according to their status (e.g. reservation set, no reservation set). By selecting the nodes more information is displayed (owner of the reservation, time until the reservation is valid). In another display mode the nodes are colored according to the date of the last history entry. By selecting the node, the appropriate history information is displayed.

In addition to the possibility of displaying sub-document related awareness information the structure editor provides features for displaying document related information. In the first prototype we included a user list display. Here the pictures of all users working on the document are displayed. The status of the users is indicated by placing them on the left or on the right side of a bar (active - inactive). The more detailed status values are displayed by overlaying the user pictures with the extra information (e.g. a note that the user is on leave until a given date).

In the applications for editing (sub-)documents awareness information is also displayed. We provide possibilities to display the work area and reservation information for the edited document parts. Additionally, one can access history information for the edited (sub-)document. For more information one can use parallel running navigation tools and awareness tools.

For further information about IRIS see [22] or the IRIS-Web pages[8].

6 Conclusion

In this article we briefly presented the human level view of distributed problem solving. From the coordination perspective we identified the provision of group awareness and workspace awareness as an important means for the support of cooperation within distributed human teams. We discussed the topic in detail and presented an application of the concepts in the area of support for distributed collaborative writing.

References

1. Agostini A., de Michelis G., Grasso M. A., Prinz W., and Syri A. Contexts, work processes, and workspaces. *Computer Supported Cooperative Work*, 5(2-3):223–250, 1996.
2. Baecker R. M., Nastos D., Posner I. R., and Mawby K. L. The user-centred iterative design of collaborative writing software. In Ashlund S., Mullet K., Henderson A., Hollnagel E., and White T., editors, *Proc. ACM Conf. on Human Factors in Computing Systems (INTERCHI'93)*, SIGCHI, pages 399–405. ACM Press, New York, NY, Apr. 1993.

[8] http://www11.informatik.tu-muenchen.de/proj/iris/

3. Bannon L. J. and Schmidt K. Cscw: Four characters in search of a context. In *Proc. 1st European Conf. on Comp. Supported Cooperative Work*, pages 358–372. Computer Sciences House, Sloug, UK, Sep. 1989.
4. Beaudouin-Lafon M. Beyond the workstation: Mediaspaces and augmented reality. In Cockton G., Draper S. W., and Weir G. R. S., editors, *Proc. HCI'94*, pages 9–18. Cambridge University Press, 1994.
5. Bellotti V. M. E. and Sellen A. Design for privacy in ubiquitous computing environments. In de Michelis G., Simone C., and Schmidt K., editors, *Proc. 3rd European Conf. on Comp. Supported Cooperative Work*, pages 77–92. Kluwer Academic Publishers, Dordrecht, Sep. 1993.
6. Borghoff U. M. and Schlichter J. H. *Rechnergestützte Gruppenarbeit — Eine Einführung in Verteilte Anwendungen*. Springer Lehrbuch. Springer Verlag, Berlin, 1995.
7. Clark H. H. and Brennan S. E. *Grounding in Communication*, In Resnick L. B., Levine J. M., and Behrend S. D., editors, *Perspectives on Socially Shared Cognition*, pages 127–149. Washington, DC: American Psychological Association, 1991.
8. Coulouris G., Dollimore J., and Kindberg T. *Distributed Systems, Concepts and Design*. Addison-Wesley, Reading, Mass. and London, 1994.
9. Crowe M. K., editor. *Cooperative Work with Multimedia*, volume 1 of *Research Reports ESPRIT, Project 6310 - MMTCA*. Springer Verlag, Berlin, 1994.
10. Dourish P. and Bellotti V. Awareness and coordination in shared workspaces. In Turner J. and Kraut R. E., editors, *Proc. Intl Conf. on Comp. Supported Cooperative Work*, pages 107–114. ACM Press, New York, NY, Oct. 1992.
11. Dourish P. and Bly S. Portholes: Supporting awareness in a distributed work group. In Bauersfeld P., Bennett J., and Lynch G., editors, *Proc. ACM Conf. on Human Factors in Computing Systems (INTERCHI'92)*, SIGCHI, pages 541–547. ACM Press, New York, NY, May 1992.
12. Ellis C. A. and Gibbs S. J. Concurrency control in groupware systems. In Clifford J., Lindsay B., and Maier D., editors, *Proc. ACM SIGMOD Intl Conf. on Management of Data*, volume 18, pages 399–407. ACM Press, New York, NY, June 1989.
13. Ellis C. A., Gibbs S. J., and Rein G. L. Groupware – some issues and experiences. *Communications of the ACM*, 34(1):38–58, Jan. 1991.
14. Fish R. S., Kraut R. E., and Leland M. D. P. Quilt: A collaborative tool for cooperative writing. In Allen R. B., editor, *Proc. ACM SIGOIS/IEEE TC-OA Conf. on Office Information Systems*, volume 9, pages 30–37, Mar. 1988.
15. Fuchs L. *17: Interest in Work Situations: Awareness Support Revisited*, In Bullock A. and Mariani J., editors, *Assessment and refinement of models of interaction*, pages 337–356. Lancaster Univ., Aug. 1995.
16. Fuchs L., Pankoke-Babatz U., and Prinz W. Supporting cooperative awareness with local event mechanisms: The groupdesk system. In Marmolin H., Sundblad Y., and Schmidt K., editors, *Proc. 4th European Conf. on Comp. Supported Cooperative Work*, pages 247–262. Kluwer Academic Publishers, Dordrecht, Sep. 1995.
17. Greenberg S. Peepholes: Low cost awareness of one's community. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems, Companion Proceedings*, pages 206–207, Apr. 1996.
18. Greenberg S., Gutwin C., and Cockburn A. Using distortion-oriented displays to support workspace awareness. Technical report, Dept of Comp. Science, Univ. of Calgary, Canada, Jan. 1996.
19. Greif I., editor. *Computer-supported cooperative work: A book of readings*. Morgan Kaufmann Publ. Inc., Los Altos, CA, 1988.

20. Johansen R. *Groupware: Computer Support for Business Teams.* The Free Press, Macmillan Inc, NY, 1988.
21. Johansen R. Teams for tomorrow. In *Proc. 24th IEEE Hawaii Intl Conf. on System Sciencs*, pages 520–534. IEEE Comp. Soc. Press, Los Alamitos, 1991.
22. Koch M. Design issues for a distributed multi-user editor. *Computer Supported Cooperative Work — An International Journal*, 3(3-4):359–378, 1995.
23. Koch M. *Unterstützung kooperativer Dokumentenbearbeitung in Weitverkehrsnetzen.* PhD thesis, Inst. für Informatik, Techn. Univ. München, Germany, 1997. (also available as: 'Kooperation bei der Dokumentenbearbeitung - Entwicklung einer Gruppeneditorumgebung für das Internet', DUV, Wiesbaden, 1997).
24. Lai K.-Y. and Malone T. W. Object lens: A "spreadsheet" for cooperative work. In *Proc. Intl Conf. on Comp. Supported Cooperative Work*, pages 115–124. ACM Press, New York, NY, Sep. 1988.
25. Lay M. M. and Karis W. M., editors. *Collaborative Writing in Industry: Investigations in Theory and Practice.* Baywood Publishing Company, Amityville, 1991.
26. Malm P. S. CSCW and Groupware, a classification of CSCW-systems in a technological perspective. Technical report, University of Tromsø, Apr. 1994. masters thesis, only available in Norwegian.
27. Malone T. W. and Crowston K. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, Mar. 1994.
28. Miles V. C., McCarthy J. C., Dix A. J., Harrison M. D., and Monk A. F. *Reviewing Designs for a Synchronous-Asynchronous Group Editing Environment*, In Sharples M., editor, *Computer Supported Collaborative Writing*, pages 137–160. Springer Verlag, London, 1993.
29. Mullender S., editor. *Distributed Systems.* Addison-Wesley, Reading, Mass. and London, 2nd edition edition, 1993.
30. Neuwirth C. M., Chandhok R., Kaufer D. S., Erion P., Morris J. H., and Miller D. Flexible diff-ing in a collaborative writing system. In Turner J. and Kraut R. E., editors, *Proc. Intl Conf. on Comp. Supported Cooperative Work*, pages 147–154. ACM Press, New York, NY, Oct. 1992.
31. Posner I. R. and Baecker R. M. *How people write together*, In Baecker R. M., editor, *Readings in groupware and computer- supported cooperative work, assisting human-human collaboration.* Morgan Kaufmann Publ. Inc., Los Altos, CA, 1993.
32. Rodden T. *17: Objects in Space, the Spatial Model and Shared Graphs*, In Benford S., Bullock A., Fuchs L., and Mariani J., editors, *Computable Models and Prototypes of Interaction*, pages 355–379. Lancaster Univ., Oct. 1994.
33. Rodden T. Population the application: A model of awareness for cooperative applications. In *Proc. Intl Conf. on Comp. Supported Cooperative Work*, pages 87–96, Nov. 1996.
34. Smith R. G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. on Computers*, C-29(12):1104–1113, Dec. 1980.
35. Tanenbaum A. S. *Computer Networks - third edition.* Prentice Hall, Upper Saddle River, NJ, 1996.
36. Wilson P. *Computer Supported Cooperative Work.* Oxford, UK: Intellect Books, 1991.
37. Wulf V. and Hartmann A. *The Ambivalence of Network Visibility in an Organizational Context*, In Clement A., Kolm P., and Wagner I., editors, *NetWorking: Connecting Workers In and Between Organizations*, pages 143–152. North Holland, Amsterdam, 1994.