

Putting it all together: Requirements for a CSCW platform

Kjeld Schmidt

Cognitive Systems Group
Risø National Laboratory
DK-4000 Roskilde, Denmark
kschmidt@risoe.dk

Tom Rodden

Computing Department
Lancaster University
Lancaster LA1 4YR, UK
tom@comp.lancs.ac.uk

Abstract. CSCW systems have generally failed to meet the requirements of users in actual cooperative work settings, primarily due to constraints imposed by current platform architectures that do not adequately support the fluent transitions between formal and informal interaction or the inextricable interweaving of individual and cooperative work that characterizes everyday work practice. Based on a sociological conceptualization of cooperative work, the paper outlines the requirements for a CSCW platform that is characterized by a clear division of labour between CSCW applications that incorporate domain-specific mechanisms of interaction and a CSCW platform providing a set of generic techniques of communication accessible to CSCW and single-user applications alike.

In spite of its recent inception, CSCW has come a long way. An array of very promising technologies is being developed and explored. As yet, however, only a handful of software products have proved to be commercially viable. In fact, CSCW systems have generally failed to meet the requirements of users in actual cooperative work settings. They do not adequately support the fluent transitions between formal and informal interaction or the inextricable interweaving of individual and cooperative work that characterizes everyday work practice. To a large extent, this deficiency can be attributed to fundamental conceptual problems in the 'architecture' of current platforms. It is becoming increasingly clear that current operating systems are inadequate as platforms for CSCW systems. Actually, the very issue of what should be conceived of as CSCW applications - as opposed to facilities of a CSCW operating system - is wide open. What is the allocation of CSCW functions between the operating system and applications or between CSCW and non-CSCW facilities?

The operating systems of workstation platforms have been some 20 years in the making and the current allocation of function between operating system and applications is the result of an evolutionary process. In the course of time, system developers repeatedly realized that they were replicating code and that it would be more efficient to include certain functions in the operating system for several applications to use. Simultaneously, users experienced the hassles and traumas of working with self-contained applications that did not support the fluent shift between tasks and, hence, did not deliver the expected productivity gains. Slowly a wide ar-

ray of functions were perceived to be generic and they were then assigned to the operating system. The CSCW field has not experienced 20 years of evolutionary development, of course. But in view of the radical way in which CSCW facilities intervene in how people work together and the potentially disruptive effects such systems therefore may have, it is unlikely that users will accept decades of trial-and-error experiments. We simply have to do better than that for CSCW systems to be acceptable.

In the following, we will examine the relationship between formal and informal activities as well as cooperative and individual activities in cooperative work and, based on that, outline the requirements for a CSCW platform.

Cooperative work and articulation work

Work is always immediately social in the sense that the object and the subject, the ends and the means, the motives and the needs, the implements and the competencies, are socially mediated. Moreover, the very work process often involves multiple people that are *mutually dependent* in their work and therefore are required to *cooperate* in order to get the work done (Schmidt, 1991b).

The notion of mutual dependence *in work* does not refer to the interdependence that arises simply having to share the same resource. They certainly have to coordinate their activities but to each of them the existence of the others is a mere nuisance and the less their own work is affected by others the better. The time-sharing facilities of operating systems for mainframe computers cater for just that by making the presence of other users imperceptible. Being mutually dependent *in work* means that 'A' relies positively on the quality and timeliness of 'B's work and vice versa and should primarily be conceived of as a positive, though by no means necessarily harmonious, interdependence.

Because of this interdependence, cooperating workers have to articulate (divide, allocate, coordinate, schedule, mesh, interrelate, etc.) their distributed individual activities (Strauss, 1985; Gerson and Star, 1986). Thus, by entering into cooperative work relations, the participants must engage in activities that are, in a sense, extraneous to the activities that contribute directly to fashioning the product or service and meeting the needs. That is, compared with individual work, cooperative work implies an overhead cost in terms of labor, resources, time, etc. The obvious justification for incurring this overhead cost and thus the reason for the emergence of cooperative work formations is, of course, that workers could not accomplish the task in question if they were to do it individually (Schmidt, 1990). In order to be able to articulate the distributed activities of a cooperative work arrangement, the participants need access to appropriate means of communication.

What is appropriate naturally depends on the specific characteristics of the cooperative work arrangement. In this context, the popular two-dimensional taxonomy of CSCW facilities (e.g., Johansen, 1988, p. 44) is not very helpful:

	Co-located	Remote
Synchronous		
Asynchronous		

On one hand, the distinction between co-located and remote cooperation is a category mistake. Distance in space is only an issue in so far as the techniques of communication available for interaction across that space offers restricted bandwidth and delayed feedback compared with the full capacity of face-to-face interaction. By providing techniques of communication in the form of file sharing, shared view, email, computer conferencing, and video conferencing that increase the bandwidth and reduce the turnaround time of communication, CSCW systems can augment the capacity of cooperating ensembles of articulating their distributed activities in spite of distance. On the other hand, the distinction between synchronous and asynchronous interaction may be meaningful in so far as it represents the important phenomenon of feedback delay or turnaround time as determined by the different techniques of communication. The distinction is not quite appropriate, however. It conceives of CSCW facilities in terms of the characteristics of the medium as opposed to the characteristics and requirements of the the cooperative effort. And in addition, it creates a dichotomy where a scale is more appropriate.

In contrast, CSCW facilities should be categorized in terms of requirements rather than characteristics of the medium. If we do this, different pertinent characteristics of cooperative work can be highlighted as being of prime importance for the requirements for the techniques of communication:

- The degree and nature of interdependence between members of the cooperating ensemble as determined by the field of work, e.g. the rigorous and causal coupling of different functions of a power plant *versus* a loose and indeterministic interdependence of the distributed activities within a research area.
- The extent to which the field of work requires instantaneous reactions to events and, hence, rapid articulation of activities, e.g., the time critical demand on air traffic control *versus* the more pedestrian pace allowed for processing tax cases.
- The extent to which the field of work is characterized by incomplete, ambiguous, erroneous, and contradictory information; by incomplete, equivocal, contradictory, or ephemeral criteria; by a conceptual world of a rich and varied semantics, etc., and, accordingly, the extent to which tasks involve discretionary decision making and concomitant negotiations, e.g., the relatively unambiguous character of ticket reservation *versus* the decidedly discretionary character of legal proceedings, medical diagnosis, policy making, etc.

These dimensions of cooperative work can be translated into requirements for the techniques of communication in terms of bandwidth (bits per second) and turnaround time (the delay from dispatch to reply as determined by the medium).

For example, a low degree of interdependence of distributed activities and time pressure on their execution and articulation allows for more sporadic interaction by means of channels of communication with a high turnaround time, say, surface mail. On the other hand, a high degree of interdependence and time pressure requires a permanently open channel of communication with minimal turnaround time so as to convey the multitude of inconspicuous cues that are required for cooperators to acquire and maintain peripheral or general awareness of the changing state of affairs within and beyond the cooperating ensemble (cf., e.g., Gaver, 1991; Heath and Luff, 1991).

Likewise, articulation of distributed activities that involve discretionary decision making will typically require, at least intermittently, various negotiation processes. For this purpose, conventional co-located ‘face-to-face’ interactions provide the required large bandwidth, not only in terms of gigabits per second but also, and more importantly, in terms of a rich variety of interactional modalities (grunts, intonations, facial expressions, gestures, gesticulations) with powerful and flexible social connotations.

It should be noted, however, that the requirements of articulation work for the techniques of communication are neither stable nor consistent. On one hand, different tasks within the same cooperative work setting may require support by different media, and any one person within the ensemble may be involved in multiple tasks, over time or concurrently. Also, the time pressure under which a given task or class of tasks is carried out may change over time. Consequently, cooperating workers will typically be changing medium quite frequently and for a given task they will typically be applying an arsenal of media and modalities, ‘synchronous’ as well as ‘asynchronous’ (Bignoli et al., 1991). In a meeting, for example, participants will use the agendas that may have been distributed in advance; they may bring files from archives as well as prepared notes and overhead transparencies; in the course of the negotiations they may use whiteboards, flipovers, etc. to convey suggestions, organize the debate, and retain results achieved; they will take notes and minutes for future use; and in all this they will mobilize the powerful repertoire of human speech and body language. All more or less concurrently and meshed in a very fluent way.

Now, should these techniques of communication be conceived of and implemented as CSCW applications or as CSCW functions of the operating system? Since these techniques of communication are semantically neutral in the sense that they can be applied in articulation work in all work domains, we will argue that these techniques of communication should be conceived of as functions of the operating system. In order to clarify the argument, let us first look at the category of functionality that constitutes the core of CSCW applications in the proper sense of the word.

Mechanisms of interaction in CSCW

As observed above, cooperating workers need to articulate their partial activities. In most of the CSCW oriented research thus far, it has - implicitly or explicitly - been the underlying assumption that the cooperative work arrangement to be supported by a computer artifact is a small, stable, egalitarian, homogeneous, and harmonious ensemble of people, a 'group'. This assumption is not tenable (Bannon and Schmidt, 1989). Cooperative work in natural settings, has a number of characteristics that must be taken into account if CSCW systems are to be acceptable to users and, hence, commercially viable:

- Cooperative ensembles are either large, or they are embedded within larger ensembles.
- Cooperative ensembles are often transient formations, emerging to handle a particular situation after which they dissolve again.
- Membership of cooperative ensembles is not stable and often even non-determinable. Cooperative ensembles typically intersect.
- The pattern of interaction in cooperative work changes dynamically with the requirements and constraints of the situation.
- Cooperative work is distributed physically, in time and space.
- Cooperative work is distributed logically, in terms of control, in the sense that agents are semi-autonomous in their partial work. Cooperative work involves incommensurate perspectives (professions, specialties, work functions, responsibilities) as well as incongruent strategies and discordant motives.
- There are no omniscient agents in cooperative work in natural settings.

In 'real world' settings characterized by dispersed, distributed, and dynamic cooperative work arrangements and involving a large, varying, or indeterminate number of participants, the intuitive modes of articulation work of everyday social life do not cope effectively. Hence, articulation work becomes extremely complex and demanding. In these settings, people engaged in cooperative work arrangements apply various *mechanisms of interaction* so as to reduce the complexity and, hence, the overhead cost of articulation work, e.g.:

- Organizational structures in the form of formal (explicit, statutory, legally enforceable) and not so formal (implicit, traditional, customary) allocation of resources, rights, and responsibilities within the cooperating ensemble.
- Plans, schedules, e.g. master schedules and kanban systems in manufacturing enterprises (Schmidt, 1991a).
- Standard operating procedures (Suchman, 1983; Suchman and Wynn, 1984).
- Conceptual schemes (e.g., thesauruses, taxonomies) for indexation or classification of information objects so as to organize distributed inclusion and retrieval of objects in 'public' repositories, archives, libraries, databases etc. maintained by multiple persons.

These protocols, formal structures, plans, procedures, and schemes can be conceived of as *mechanisms* in the sense that they (1) are objectified in some way (explicitly stated, represented in material form), and (2) are deterministic or at least give reasonably predictable results if applied properly. And they are *mechanisms of interaction* in the sense that they reduce the complexity of articulating cooperative work.

In various ways, these mechanisms embody representations of important aspects of the field of work of the cooperating ensemble. They relate to a specific work domain. Conceptual schemes, for example, are obviously domain specific in that they represent a model of the conceptual structure of a domain. The same applies to organizational structures. Also, the master schedule of production control systems in manufacturing embodies an elaborate model of the products, their components (Bill of Materials), the average production or purchasing time and cost of each component or subassembly, etc. Likewise, an organizational procedure refers to typical classes of cases characteristic of the particular domain at hand. It may convey information on the functional requirements to be met by the process and the product; it may highlight decisional criteria of crucial import; it may suggest a strategy for dealing with a specific type of problems; it may indicate pitfalls to avoid; or it may simply provide an *aide memoir*. And, finally, it may express some statutory constraints. In short, mechanisms of interaction like the ones mentioned above are enmeshed in the semantics of the particular domain.

A major research effort in CSCW has been directed at incorporating such mechanisms of interaction in CSCW systems:

- Plans, schedules, e.g. calendar systems, THE COORDINATOR (Flores et al., 1988).
- Standard operating procedures, e.g. work flow management applications such as WORKHORSE, DOMINO (Kreifelts et al., 1991), STRUDL (Shephard et al., 1990).
- Organizational structures, e.g. AMIGO and COSMOS (Benford, 1988; Bowers et al., 1988; Danielsen and Pankoke-Babatz, 1988; COSMOS, 1989), and a computational organizational model such as that suggested by the MOCCA project (Benford, 1992).
- Conceptual schemes, e.g. classification schemes, taxonomies, and thesauri for reactive databases and hypermedia systems such as the Community Handbook proposed by Engelbart and Lehtman (Engelbart and Lehtman, 1988).

As opposed to CSCW facilities that merely or primarily provide a channel of communication, CSCW facilities that incorporate a mechanism of interaction can be seen as a distinct category of applications in the sense that they are semantically biased. These are genuine *CSCW applications*. This point is by no means merely terminological.

Precisely because of the dynamic and distributed character of cooperative work arrangements, mechanisms of interaction are not executable code but rather heuristic

and vague statements to be interpreted and instantiated, maybe even by means of intelligent improvisation. They are local and temporary closures with a limited area of validity and they are by necessity underspecified. As pointed out by Suchman (Suchman, 1987, p. 52):

“plans are resources for situated action, but do not in any strong sense determine its course. While plans presuppose the embodied practices and changing circumstances of situated action, the efficiency of plans as representations comes precisely from the fact that they do not represent those practices and circumstances in all of their concrete detail.”

This observation applies to mechanisms of interaction in general. To be made to work, they themselves need to be managed, i.e., constructed, maintained, developed, interpreted, applied, adapted, circumvented, modified, executed, represented, and negotiated. This secondary level of articulation work is, of course, also performed cooperatively (Schmidt, 1991b).

In order to support the dynamic integration of formal and informal cooperative work, a CSCW system should make the incorporated mechanism of interaction accessible to users and, indeed, support users in interpreting the mechanism and evaluating its rationale and implications. It should support users in applying and adapting the mechanism to the situation at hand; i.e., it should allow users to tamper with the way it is instantiated in the current situation, execute it or circumvent it, etc. The system should even support users in modifying the underlying mechanism and in creating new mechanisms in accordance with the changing organizational realities and needs. However, since the management of mechanisms of interaction is itself a cooperative activity, the system should support the documentation and communication of decisions to apply, adapt, modify, circumvent, execute, etc. the underlying mechanism. And, last but not least, in all of this the CSCW system as a whole should support the process of negotiating the interpretation, application, adaptation, modification, circumvention, execution etc. of the mechanisms of interaction incorporated in the system.

In order to facilitate the cooperative management of mechanisms of interaction and especially the negotiations required in this management activity, CSCW applications, i.e. software facilities incorporating domain-related mechanisms of interaction, need to be supported by and combined with a generic set of relatively unrestricted techniques of communication of a requisite bandwidth and turnaround time (Robinson, 1991). The distinction between semantically prejudiced mechanisms of interaction and semantically neutral techniques of communication thus provides a general criterion for the allocation of functions between CSCW applications and operating system. This general distinction also provides a basis for supporting the fluid integration of cooperative work and individual work.

Cooperative work and individual work

The relationship between cooperative work and individual work is quite tricky.

We are indeed social animals, but we are not *all* of us *always* and in *every* respect mutually dependent in our work. Thus, in spite of its intrinsically social nature, work is not intrinsically cooperative in the sense that workers are mutually dependent in their work. Cooperative work is thus distinct from individual work, in theory as well as in practice. In so far as cooperating workers have to articulate their distributed individual activities and, thus, must engage in activities that are extraneous to the activities that contribute directly to fashioning the product or service and meeting the needs, cooperative work has characteristics distinctly different from individual work.

Yet, cooperative work and individual work should not be conceived of as different work domains. The category ‘cooperative work’ is orthogonal to ‘individual work’. Cooperative work pertains to the *same work domain* as individual work: writing, editing, designing, engineering, etc. But it is a *different way* of doing *the same*.

In daily work practice, cooperative and individual activities are inextricably interwoven. Cooperative work is always conducted by individuals (albeit interdependently and hence concertedly), and yet, individual activities are always penetrated and saturated by cooperative work as by a social ‘ether’ (Hughes et al., 1991). More than that, the boundary between individual and cooperative work is dynamic in the sense that people enter into cooperative work relations and leave them according to the requirements of the current situation and the technical and human resources at hand. That is, cooperative work arrangements emerge contingently, to dissolve again into individual work. Cooperative work is punctuated by individual work and vice versa. Over time, people shift between individual and cooperative activities and, while engaged in cooperative work activities, they may be simultaneously involved in parallel streams of activity conducted individually. In real life, it is all a mess.

A CSCW system should support the fluent meshing of individual work and cooperative work. In all its generality, this statement may seem uncontroversial. Nevertheless, most CSCW products do not support this fluency. For example, when composing an email message the user should not be required to shift to a special editor, that is, for example, required to leave the word processor normally used for composing letters, writing reports etc. The same applies to CSCW facilities supporting cooperative authoring, conferencing, etc. The commercial groupware product ASPECTS, for example, allows multiple users to cooperate on writing a document. However, they are required to shift to the word processing facility of ASPECTS in order to cooperate. The effect of this that the system creates an impedance between cooperative and individual activities.

As argued above, CSCW facilities that support cooperative work by increasing the bandwidth of the communication channel or by reducing the turnaround time should not be conceived of as applications but as *operating system functions* accessible to the appropriate applications. If they are not conceived of *and implemented* as system functions that can be accessed from and combined with applications, the delicate and dynamic relationship between cooperative and

individual work breaks down. This applies to traditional single-user applications as well as genuine CSCW applications.

Another way of supporting the fluent integration of cooperative and individual activities is, of course, to provide a software package with a selection of integrated facilities (word processing, email, file retrieval, etc.). LOTUS NOTES is an outstanding example of this strategy. The problem with this strategy is as old as the strategy itself, however. The integration is ensured by ad hoc interfaces between different facilities as opposed to standardized interfaces (standard formats, Cut and Paste functions, Publish and Subscribe functions, etc.) that are accessible to any application. In the style of SYMPHONY, JAZZ, MICROSOFT WORKS, users are prescribed a selection of middle-of-the-road facilities. As a philosophy for a CSCW platform this strategy implies that all users of the system are forced to use the same middle-of-the-road facilities. The strategy thus amounts to abandoning the open architectures of modern platforms and returning to the regime of the mainframe age.

In sum, in order to support the fluid interweaving of informal and formal activities as well as cooperative and individual activities, the allocation of function between general OS facilities and specific applications should be planned carefully.

Requirements for a CSCW platform

Based on a sociological conceptualization of the situated nature of cooperative work and its articulation, the previous sections have argued for a clear functional division of labor between the CSCW platform and the various applications.

The dynamic character of cooperative work and its articulation is such that a focus should be placed on a computational platform that provides a range of support facilities for CSCW applications.

CSCW facilities vary greatly and a number of researchers have investigated how these characteristics are related (Ellis et al., 1991; Rodden, 1991). The most popular characteristics used to distinguish CSCW facilities have been the *distance* between cooperating users and the *time* dependency involved in the interaction taking place (Johansen, 1988). In contrast, we have argued for a distinction between applications incorporating various domain-specific mechanisms of interaction and a repertoire of universally applicable techniques of communication.

We would characterize CSCW applications as being computational components that incorporate various mechanisms of interaction which in some form represent the semantics of the domain of the cooperating ensemble. The type of semantics captured and the techniques of representation used vary greatly from highly structured systems based on speech act theory (Winograd and Flores, 1986) to less structured brainstorming applications, for example, the COGNOTER system (Stefik et al., 1987).

These applications make use of or may be combined with a wide range of techniques of communication to enable unrestricted interaction between users. It is our belief that these techniques of communication can be collected to form a semanti-

cally neutral CSCW platform to support CSCW applications as well as single-user applications. This section examines the nature of the generic facilities to be provided by a CSCW platform and the properties such a platform should exhibit.

Supporting informal interaction

To support the ‘informal’ interaction required to articulate distributed activities in dynamic and complex settings, a CSCW platform should provide facilities for unrestricted interaction among users in the form of channels of direct communication. The informal nature of the communication needed is such that these channels will require a significant bandwidth and very low turnaround time and may exploit multimedia facilities such as audio and video links (Ishii, 1990). Such communication facilities could be conceived of as a **Conference** function.

Facilities of this form are currently provided by CSCW applications such as CRUISER (Root, 1988). However, these informal interaction facilities should be provided independently of specific applications. For example, a **Conference** function might be used to support the ongoing negotiation required to coordinate the distributed activities of several users working on the same single-user application. Or it can support the cooperative management of a mechanism of interaction incorporated in a CSCW application. This approach is adopted within the on-going MOCCA project (Benford, 1992) which is developing an architectural model for an environment to support CSCW applications.

Supporting information sharing and exchange

A wide range of cooperative tasks requires the exchange and sharing of particular information objects. The platform should thus provide a set of CSCW functions that are accessible from any application and can be executed on any object, irrespective of the specific application by means of which the particular object has been created. Given the flexible and complex nature of cooperative work, a spectrum of sharing facilities should be provided, ranging from a narrow to a very large bandwidth and from the loose to the tightly coupled interaction, e.g.:

Exchange Object. A CSCW platform should enable users to exchange (send and receive copies of) information objects created by any application. By providing this facility, a CSCW platform would allow users to apply their preferred ‘single user’ applications (e.g., word processors) in cooperative activities. This facility will obviously exploit an underlying electronic message service such as X.400 (CCITT, 1988).

Share Object. A CSCW platform should enable users to share an information object, i.e. to make a particular object (e.g., a file) accessible to other users. As in the case of information exchange, this facility should be accessible to users from any application. This is analogous to the sharing facility provided under the Macintosh System 7 (Apple Computer, 1991). However, these facilities should be extended to support the management of dynamic groups

within a distributed setting. They should also be provided in such a way that users are aware of the access by other users. This contrasts with the highly transparent approach adopted by traditional multi-user database systems (Mariani and Rodden, 1991).

Share View. A CSCW platform should enable users to work in close interaction by giving other users access a particular application window (Greenberg, 1990). Given the need to support a range of applications the platform should allow both *cooperation aware* and *cooperation transparent* window sharing (Lauwers and Lantz, 1990). Cooperation transparent facilities will allow windows to be selected and designated as shared windows by users. This technique is currently exploited by systems such as SHARED_X (Gust, 1988) and MMCONF (Crowley et al., 1990). In addition a set of window sharing primitives should be provided to CSCW application developers in order to encourage the development of cooperation aware applications within the platform.

In order to support the fluid transitions between individual and cooperative activities, these function should work much like a **Print** command, i.e. as a ubiquitous system command that performs certain operations on a selected object controlled by a specific application. In other words, the user should be able to publish a file, send a message, display a window, etc. to another user just as he or she can print it from an ordinary workstation.

Supporting decision making

The need to reach agreement on a particular issue occurs sufficiently often and within such a diversity of settings to suggest that it is independent of the task taking place. A CSCW platform may thus be required to provide facilities for supporting a distributed decision making process. The technique should at least provide the ability for a number of disparate group members to vote on a range of issues. This *voting* mechanism should be provided in a policy free manner and allow a set of different voting procedures to be used to drive the voting mechanism. The semantics of any decision being made would not be of interest to the platform but may be of concern to applications or users.

A **Vote** function could for instance be applied to make a decision among a large ensemble of perhaps widely distributed actors to use a specific turntaking protocol in a shared view session, to change a particular organizational procedure temporarily or permanently, or to modify the domain specific indexation scheme.

Supporting coordination and control protocols

While mechanisms of interaction that reduce the complexity of articulation work typically embody domain-specific semantics of no concern to other application domains and therefore of no concern to the operating system, a few mechanisms are

indeed of general utility. They are therefore important to the provision of general services within a CSCW platform, e.g.:

Turntaking Protocol. Generic CSCW facilities such as Conference, Share Object, Share View that provide relatively unrestricted techniques of communication normally require some form of turntaking protocol to coordinate floor control, i.e. to manage the allocation of control of the session. A CSCW platform should provide such techniques in such a manner that a range of different turntaking protocols can be applied. These protocols should be accessible to users who should be able to tailor them to meet particular needs. An example of this approach to floor control is provided by the SHARE system developed by (Greenberg, 1991).

Sharing and Access Policy. Existing computer platforms embody a wide range of design decisions and assumptions about their expected use. These policy decisions are seldom made explicit within systems, yet they often characterize behavior. This is most evident within distributed systems where the policies concerning distribution transparency are often unsuitable for the needs of CSCW applications (Rodden and Blair, 1991). A number of researchers within the distributed systems community have begun to examine the role of policy within systems. In particular, researchers have suggested the need for a policy object (Moffet and Sloman, 1991) which allows policies to be explicitly represented within systems. A similar approach will be important for a CSCW platform. This policy information needs to be recorded in such a way it can readily accessed and understood by cooperating users.

Supporting domain directories

System functions such as the Conference, Send Object, Share Object, and Share View functions and generic coordination mechanisms such as the Turntaking and Transparency protocols require integrated support from a set of domain models.

First, a **User Directory** is required to provide the system functions of the CSCW platform with data pertaining to users and their organizational relationships. While the User Directory is organization specific, it is nevertheless generic to a particular organizational setting in the sense that it provides data that is common to a wide array of specific facilities and applications: User names, access codes, group membership, group administrator names, etc. User details could be held within a distributed register (for example, that provided by X.500 (ISO and CCITT, 1988)). A User Directory would also provide basic support for various CSCW applications such as work flow management systems and organizational structure models.

Second, an **Object Directory** is required to support the distributed indexation of objects so that they can be retrieved with reasonable ease and certainty by other users. The Object Directory should at least record names of the creator of an object, the time of creation, names and times of subsequent changes to the object, relations to other objects (copy of, comment on, reply to, etc.). In addition, the Object Directory should provide tools for building a domain-specific classification scheme

that defines the relations between categories (e.g., a thesaurus). The purpose of such a classification scheme would be to support domain-related indexing of information objects so as to support the distributed inclusion and retrieval of public objects.

Conclusion

The requirements for a CSCW platform outlined in this paper originate from, and bring together, two different approaches to CSCW and two separate strands of CSCW research. The paper has attempted to provide a mapping between sociological conceptions of real-world cooperative work and software engineering conceptions of the functionality and architecture of a CSCW platform such as those proposed by the MOCCA project (Benford, 1992). The paper has demonstrated how these two separate strands of research have arrived at a similar set of results.

Acknowledgements

The paper has resulted from on-going discussions within and between a number of European interdisciplinary projects examining both theoretical and technical aspects of CSCW. We have especially benefited from the research activities carried out under the auspices of CoTech WG2 (MOCCA) developing an environment to support CSCW applications and CoTech WG4 (CSCW Design Concepts) examining the conceptual basis for the design of CSCW systems, both of which have been supported by a grant from the ESPRIT Basic Research program of the Commission of the European Community.

The paper also benefitted from the initial work in the ESPRIT Basic Research action COMIC (“Computer-based Mechanisms of Interaction in Cooperative Work”).

Among the many people who have been involved in these investigations of CSCW systems requirements, we are particularly indebted to Liam Bannon, Steve Benford, Leandro Navarro, Wolfgang Prinz, Mike Robinson, and Dan Shapiro.

References

- Apple Computer: *Macintosh™ Networking Reference. System 7*, Apple Computer, Inc., Cupertino, CA, 1991.
- Bannon, Liam, and Kjeld Schmidt: “CSCW: Four Characters in Search of a Context,” *EC-CSCW '89. Proceedings of the First European Conference on Computer Supported Cooperative Work, Gatwick, London, 13-15 September, 1989*, 1989.
- Benford, Steve: “A data model for the AMIGO communication environment,” in *Research into Networks and Distributed Applications. European Teleinformatics Conference - EUTECO '88 on Research into Research into Networks and Distributed Applications, Vienna, Austria, April 20-22, 1988*, edited by R. Speth, North-Holland, Brussels and Luxemburg, 1988, pp. 97-109.
- Benford, Steve: “MOCCA - A CSCW Environment,” Submitted for CSCW '92, Toronto, November 1-4, 1992.

- Bignoli, Celsina, Giorgio De Michelis, and Renata Tinini: "UTUCS: A support for synchronous and asynchronous communication," *International Workshop on CSCW, Berlin, April 9-11, 1991*, edited by K. Gorling and C. Sattler, Institut für Informatik und Rechentechnik, 1991, pp. 74-84.
- Bowers, John, John Churcher, and Tim Roberts: "Structuring computer-mediated communication in COSMOS," in *Research into Networks and Distributed Applications. European Teleinformatics Conference - EUTEKO '88 on Research into Research into Networks and Distributed Applications, Vienna, Austria, April 20-22, 1988*, edited by R. Speth, North-Holland, Brussels and Luxemburg, 1988, pp. 195-209.
- CCITT: *Message Handling Systems: System Model - Service Elements, Recommendation X.400*, 1988.
- COSMOS: *Specification for a Configurable, Structured Message System*, Cosmos Coordinator's Office, Queen Mary College, 1989. [68.4 Ext/ALV].
- Crowley, T. et al.: "MMConf: An infrastructure for building shared multimedia applications," in *Proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990*, ACM press, New York, N.Y., 1990.
- Danielsen, Thore, and Uta Pankoke-Babatz: "The AMIGO Activity Model," in *Research into Networks and Distributed Applications. European Teleinformatics Conference - EUTEKO '88 on Research into Research into Networks and Distributed Applications, Vienna, Austria, April 20-22, 1988*, edited by R. Speth, North-Holland, Brussels and Luxemburg, 1988, pp. 227-239.
- Ellis, C. A., S. J. Gibbs, and G. L. Rein: "Groupware: Some issues and experiences," *Communications of the ACM*, vol. 34, no. 1, January 1991, pp. 38-58.
- Engelbart, Douglas, and Harvey Lehtman: "Working together," *Byte*, December 1988, pp. 245-252.
- Flores, Fernando et al.: "Computer Systems and the Design of Organizational Interaction," *ACM Transactions on Office Information Systems*, vol. 6, no. 2, April 1988, pp. 153-172.
- Gaver, William W.: "Sound Support for Collaboration," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 293-308.
- Gerson, Elihu M., and Susan Leigh Star: "Analyzing Due Process in the Workplace," *ACM Transactions on Office Information Systems*, vol. 4, no. 3, July 1986, pp. 257-270.
- Greenberg, Saul: "Sharing views and interactions with single-user applications," *COIS '90: Proceedings of the Conference on Office Information Systems, Boston, April 1990*, 1990.
- Greenberg, Saul: "Personalizable groupware: Accomodating individual roles and group differences," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 17-31.
- Gust, P.: "Shared X: X in a distributed group work environment," *The 2nd Annual X conference, M.I.T, Boston, January 1988*, 1988.
- Heath, Christian, and Paul Luff: "Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 65-80.
- Hughes, John, Dave Randall, and Dan Shapiro: "CSCW: Discipline or Paradigm? A sociological perspective," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 309-323.

- Ishii, Hiroshi: "TeamWorkStation: Towards a Seamless Shared Workspace," *CSCW 90, Los Angeles, CA, October 7-10 1990, New York, N.Y.*, ACM press, 1990, pp. 13-26.
- ISO and CCITT: *Information Processing Systems - Open Systems Interconnection - The Directory. CCITT X.500-X.521*, 1988. [ISO 9594-1-8].
- Johansen, Robert: *Groupware. Computer Support for Business Teams*, The Free Press, New York and London, 1988.
- Kreifelts, Thomas et al.: "Experiences with the DOMINO Office Procedure System," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 117-130.
- Lauwers, J. C., and K. A. Lantz: "Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems," *Proceedings of CHI '90, Seattle, Washington, April 1-5, 1990*, ACM, 1990, pp. 303-311.
- Mariani, John A., and Tom A. Rodden: "The Impact of CSCW on Database Technology," *International Workshop on CSCW, Berlin, April 9-11, 1991*, edited by K. Gorling and C. Sattler, Institut für Informatik und Rechentechnik, 1991, pp. 146-161.
- Moffet, J., and M. Sloman: "The Representation of Policies as System Objects," *Proceedings of the Conference on Organisational Computer Systems (COCS '91), Atlanta, Georgia, 5-8 November, 1991*, 1991.
- Robinson, Mike: "Double level languages and co-operative working," *AI & Society*, vol. 5, 1991, pp. 34-60.
- Rodden, Tom: "A Survey of CSCW Systems," *Interacting with Computers*, vol. 3, no. 3, 1991, pp. 319-353.
- Rodden, Tom, and Gordon Blair: "CSCW and Distributed Systems: The Problem of Control," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991, pp. 49-64.
- Root, R. W.: "Design of a Multi-Media Vehicle for Social Browsing," *CSCW '88. Proceedings of the Conference on Computer-Supported Cooperative Work, Portland, Oregon, September 26-28, 1988*, ACM, New York, N. Y., 1988, pp. 25-38.
- Schmidt, Kjeld: *Analysis of Cooperative Work. A Conceptual Framework*, Risø National Laboratory, Risø National Laboratory, DK-4000 Roskilde, Denmark, 1990. [Risø-M-2890].
- Schmidt, Kjeld: "Computer Support for Cooperative Work in Advanced Manufacturing," *International Journal of Human Factors in Manufacturing*, vol. 1, no. 4, October 1991a, pp. 303-320.
- Schmidt, Kjeld: "Riding a Tiger, or Computer Supported Cooperative Work," in *ECSCW '91. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, edited by L. Bannon, M. Robinson and K. Schmidt, Kluwer Academic Publishers, Amsterdam, 1991b, pp. 1-16.
- Shephard, Allan, Niels Mayer, and Allan Kuchinsky: *STRUDL - An Extensible Electronic Conversation Toolkit*, Hewlett-Packard Laboratories, 1990. [HPL-90-100].
- Stefik, M. et al.: "Beyond the chalkboard: Computer support for collaboration and problem solving in meetings," *Communications of the ACM*, vol. 30, no. 1, January 1987, pp. 32-47.
- Strauss, Anselm: "Work and the Division of Labor," *The Sociological Quarterly*, vol. 26, no. 1, 1985, pp. 1-19.
- Suchman, Lucy A.: "Office Procedures as Practical Action: Models of Work and System Design," *ACM Transactions on Office Information Systems*, vol. 1, no. 4, October 1983, pp. 320-328.
- Suchman, Lucy A.: *Plans and situated actions. The problem of human-machine communication*, Cambridge University Press, Cambridge etc., 1987.

Suchman, Lucy A., and Eleanor Wynn: "Procedures and Problems in the Office," *Office: Technology and People*, vol. 2, 1984, pp. 133-154.

Winograd, Terry, and Fernando Flores: *Understanding Computers and Cognition: A New Foundation for Design*, Ablex Publishing Corp., Norwood, New Jersey, 1986.