

Computer-Supported Cooperative Work - Concepts and Trends

Michael Koch¹, Tom Gross²

¹ Fakultät fuer Informatik
Technische Universität München
Boltzmannstr. 3
D-85748 Garching, Germany
michael.koch@acm.org

² Faculty of Media
Bauhaus-University Weimar
Bauhausstr. 11
D-99423 Weimar, Germany
tom.gross@medien.uni-weimar.de

Abstract: The research field Computer-Supported Cooperative Work (CSCW) is concerned with understanding social interaction and the design, development, and evaluation of technical systems supporting social interaction in teams and communities – or in other words it is about researching the use of computer-based technology for supporting collaboration. The field was coined in the 1980th by researchers from computer science, information science and social science. In this paper we will briefly introduce CSCW – its concepts and current trends - and thereby focus on the role of Informatics in the field – from application integration to ubiquitous user interfaces.

1 Introduction

1.1 Computer-Supported Cooperative Work (CSCW)

The research field CSCW is concerned with understanding social interaction and the design, development, and evaluation of technical systems supporting social interaction in teams and communities. Many researchers in CSCW have their own definitions of CSCW. However, most are very similar. Bowers and Benford probably have the most general sight. They write that *'In its most general form, CSCW examines the possibilities and effects of technological support for humans involved in collaborative group communication and work processes'* [BB91, p. V]. Other researchers emphasize the aspect of group work or group activity in CSCW. For instance, Greif defines CSCW as *'computer-assisted coordinated activity such as communication and problem solving carried out by a group of collaborating individuals'* [Gr88, p. XI]. For Wilson 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, services and techniques' [Wi91]. Following these characterizations, we wish to emphasize that it is important to constantly consider both sides of CSCW: firstly, the technology and computer hardware and software, and secondly, group work and social phenomena.

It is important to note that CSCW is a rather interdisciplinary field where researchers from various fields contribute with "*different perspective[s] and methodology[s] for acquiring knowledge of group work and for suggesting how the group's work could be supported*" [Gr91]. For instance, computer scientists bring in their knowledge of networks, messaging services, and distributed systems; social scientists contribute their sociological and anthropologic knowledge for questions of design; and psychologists, cognitive scientists, cognitive ergonomists, and people from human-computer interaction have already collected the knowledge of human factors in computing [GT96].

1.2 Groupware

In addition to CSCW as a name for the research field there is the term "groupware" for the technical systems resulting from CSCW research and development. In contrast to traditional computer systems that are primarily designed for a single user, the major goal of a groupware system is to assist a group of users in communicating, in collaborating, and in coordinating their activities [EGR91].

Ellis and associates state that groupware are '*computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment*' [EGR91, p. 40]

Groupware can include software, hardware, services or group process support. However, the positions of researchers diverge in their view of when distributed systems end and groupware begins, i.e. what distinguished groupware from non-groupware. In our opinion the core characteristic of (successful) groupware is the non-separation or non-isolation of users from each other. Groupware explicitly provides awareness of the co-workers and their activities and does not separate the users from each other as it is common in distributed systems in general.

Greif, who coined the term CSCW with others, realized very early that in the future many software systems will have collaborative functionality and will be groupware to some extent [Gr88, p. 12]. However, she argues that CSCW as a research field will persist, because it addresses larger questions about the design and refinement of groupware. Though more and more software systems will be addressed as groupware, there is still a need for core CSCW and groupware research and for the definition of basic features or characteristics of collaboration support tools. This can be seen in analogy to HCI, where one could claim that each software system has to deal with questions user interface design. Nevertheless, HCI or human factors in computing do have their rights to exist.

1.3 Models and Challenges

The field of CSCW has introduced several models to structure the domain of social interaction and corresponding support systems and functionality (see for [BS00] for a good overview). One we consider quite important for thinking about collaboration support is the classification on among individuals in groups into:

- coexistence,
- communication,
- coordination,
- consensus and finally
- collaboration.

Strongly coupled with the first and most basic class (coexistence) again is awareness of each other and of each others activities.

Communication as a basis for all interaction has been addressed in several models, e.g. the context-oriented communication model by Misch [Mi01]. Another model covering the functional relationship between collaborators or between collaborators and the groupware system itself is the people/artefact network or cooperative work framework [Di93 p.465f]. This framework is applicable when multiple users are utilizing some tool or system to create an interaction. As shown in Figure 1, the system is mapped as a graph with nodes and arcs. Participants (users of the system) are mapped by the nodes marked P. The system facilitating the interaction is called an artifact of work.

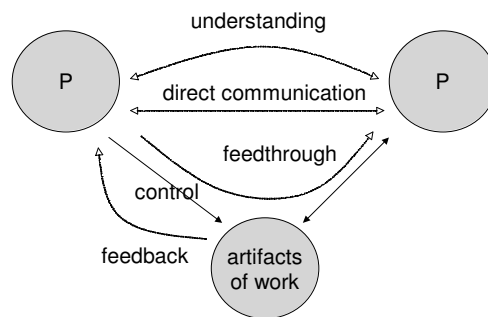


Figure 1: Communication Model [Di93]

Several authors in the field have been analysing CSCW projects and have been identifying core challenges of collaborative system design compared to software design in general. See for example the early work from Ellis or Grudin [EGR91,Gr88,Gr89].

Some of the conclusions of Grudin can be summarized in the following challenges to collaborative systems design:

- It is hard to capture the requirements for a collaborative system – 1) because so many groups and aspects need to be considered which are not intuitive to software architects, 2) the requirements usually are not clearly known to any participant, 3) the requirements or boundary conditions change over time and through the introduction of a system.
- For making a collaborative system a success all (or at least a large part) of the potential co-workers have to use the system actively (network effect, critical mass) – this requires mainly a clear balance between effort and benefit for all of the users (no disparity between effort and benefit) which also has to be communicated to the users – this includes the need for easy to use user interfaces and for a good integration.

In the following sections we address some of the core issues in the area of groupware in more detail: requirements engineering, modularization and integration,

2 Requirements Engineering and CSCW

In past CSCW development projects it often came as a surprise that the software systems developed for supporting the (distributed) groups did not work as intended. The reason for this often has been identified in the fact that the system to develop has been regarded as a technical system only, while in reality the system should have been seen as what it was, a complex sociotechnical systems – i.e. the social group that is using the system plays an important role in the success of the system. This fact, that is well recognized in information science for information systems in general, results in several problems. First, it is hard to get the requirements from the users, not only because the introduction of the system will (planned or unplanned) change the social group / organization and thereby the requirements. Second, designing a CSCW system is not only about designing the technical system, but also about shaping the social system around. And third, it still might be hard to get the potential users to accept the resulting system. Even if the system would fit their needs theoretically, the users still have to be motivated to participate in the development and the introduction of the systems. Key for solving these issues are user participation / involvement of the users and iterative / evolutionary development.

As we have already pointed out, it might be impossible to capture the final requirements at the beginning of a CSCW software project, since the requirements might change through the introduction of the system itself.

To address the problem of hard to capture requirements and of changing requirements, iterative or evolutionary development has been introduced. The basic idea behind these software development methods is not to go through the process of requirements capture, design and implementation once but several times. In the first iterations only prototypes are built and reflected back to the users to get more information on the (changing)

requirements. In Boehm's spiral model for software engineering [Bo88] several cycles are involved each of which includes the planning of the next phase, determining objectives and constraints, evaluating alternatives and resolving risks, and developing the next-level product.

CSCW research has laid special focus on how to get information from the potential users of a potential system – by watching them or by semi-structured interviews and by moderated discussions. A lot has been learned from other domains dealing with understanding social systems like ethnography. So, CSCW has taken large parts of ethnomethodology into its toolbox.

3 Technology Bundles and Application Integration

3.1 Technology Bundles

Since the domain of collaboration support is quite large – ranging from video conference support to business workflow support, in practice it is often hard to decide which functionality or tool best fits the needs of a particular situation or group. We have adapted a functional classification often used for discussing groupware (e.g. in [BS00]) for the use in practice. This classification distinguishes five application classes:

Awareness support - Awareness support is one of the core concepts in groupware. In contrast to other (multi-user) software groupware is not isolating the users from each other, but connects them, gives them a possibility to coordinate each other (implicitly). Awareness support is integrated in different tools, but also manifests in tools or libraries itself (e.g. presence awareness support with IM tools or media spaces).

Communication support - While awareness support already is a very general form of (implicit) communication, there is also need for explicit communication (support). This includes asynchronous communication (email, forums) and synchronous communication (chat, video and audio conference solutions)

Coordination support - Awareness contributes to coordination in a very general way. Nevertheless, some scenarios ask for explicit coordination support – and there are applications out there filling this need, e.g. workflow management solutions, conversation support solutions or different coordination oriented awareness solutions.

Team support - The next issue is no longer about cooperation modes like communication and coordination, but about special group types and their special needs. The first team type that values a particular application class are teams – and particular team rooms as a generic application in this domain.

Community support - In contrast to teams, communities have a completely different structure and completely different needs. Most elements from the knowledge management domain fall in this class.

Applications from these domains usually can be considered and introduced independently. Being faced with a particular situation it usually is quite straightforward to identify one or two classes that would cover the requirements, and then select appropriate tools.

3.2 CSCW Application Integration

The status of collaboration support in practice today is usually that the different functionalities are offered as separate applications – either single modules or an arbitrary collection of modules. For making collaboration support in enterprises a success there is a need to integrate the different functionalities, to integrate collaboration functionalities with standard applications, and to adapt applications to the need of the individual and the group.

Making the support tools as easy to use as possible has been one important requirement – to make sure a critical mass of users actively participates, and to ensure that potential mismatch of effort and benefit is as small as possible. One important aspect in the easiness to use is to make collaboration support easily accessible from standard work functionalities. The big problem with today's solutions in this context is that collaboration support functionality is isolated from standard work functionality. Isolated both in user interface and in data – the functionality cannot be used in an integrated way.

There are different concepts that are addressing this problem of integration. In the domain of CSCW currently the following topics are taken up:

- **Portals:** Providing collaboration functionality (especially team workspaces) as portlets to be included in standard portals – a kind of aggregating user interfaces to collaborative tools in portals – most portal vendors ship their own solutions or allow integration of other solutions via portlet standard interfaces.
- **Contextual Collaboration, Collaboration APIs and Collaboration Services:** Extending the portal idea, Microsoft launched the idea of “contextual collaboration” to integrate collaborative functionality in all standard desktop applications (to collaborate from where you need to do it). The idea involves definition of services and APIs for collaboration functionalities. Except some first examples like WEBDAV or protocols for accessing presence awareness information there are little results here.
- **Agents and Semantic Web technologies:** Personal agents, combined with semantic web technology to encode and interpret meta data is seen as an important future possibility to bring integration of applications forward. Most interface definitions already try to be based on ontologies etc.

We are addressing such issues in different projects, e.g. the development of a generic community and team support toolkit names Cobricks (see www.cobricks.org or [Ko03]), or the definition of standard architectures for collaboration support in enterprises.

4 Ubiquitous and Mobile User Interfaces for Groupware – Integration into the Workspace

In the previous section we have addressed issues of technical integration. One reason for better integration of CSCW functionality with each other and with single-user applications has been to lower the effort for the user by better integrating data and support tools, and thereby allowing a seamless (ex)change between the different tasks. Taking this idea further, the availability and modality of access to the support application can be considered a major issue.

However, support for collaborative work has been, till now, mainly determined by boundaries of stationary computers and desktop based interfaces.

Ubiquitous Computing and mobile computing, i.e. new user interfaces that are emerged in the real world, may address the boundaries of collaboration support and offer possibilities for enlarging the reach of CSCW applications. Beyond the desktop, user interfaces can be integrated more seamlessly in work processes and work environments, making access to the application and the collaborating users available wherever and whenever it is needed, anytime, anyplace and any task. Users should no longer be forced to explicitly go particular to locations for interacting with other members of the team or community.

The new possibilities (new functionalities, new user groups, new situations where collaboration support systems can be accessed) require the development of new service types and of new technological solutions. Several projects in CSCW and HCI research are currently addressing the integration of group support functionality into the workplace – often under different labels like Ubiquitous or Pervasive Computing. See [GP04,GEM06,Ko05] and [KGH02] for some examples.

Bibliography

- [Bo88] Boehm, B.W., A spiral model of software development and enhancement. *IEEE Computer* 1988. 21: p. 61-72.
- [BS00] Borghoff, U.M. and J.H. Schlichter, *Computer-Supported Cooperative Work*. 2000: Springer
- [BB91] Bowers, J. and S. Benford (eds.), *Studies in Computer-Supported Cooperative Work: Theory, Practice, and Design*. 1991, Elsevier Science: Amsterdam.
- [Di93] Dix, A.J., et al., *Human-Computer Interaction*. 1993, New York: Prentice Hall.
- [EGR91] Ellis, C.A., S.J. Gibbs, and G.L. Rein, Groupware - Some Issues and Experiences. *Communications of the ACM*, 1991. 34(1): p. 38-58.
- [Gr91] Greenberg, S., *Computer Supported Cooperative Work and Groupware*. 1991, London: Academic Press.
- [Gr88] Greif, I., *Computer-supported cooperative work: A book of readings*. 1988: Morgan Kaufmann, Los Altos.
- [GEM06] Gross, T., Eglar, T. and Marquardt, N. Sens-ation: A Service-Oriented Platform for Developing Sensor-Based Infrastructures. *International Journal of Internet Protocol Technology (IJIPT)* 1, 3 (2006). pp. 159-167.

- [GP04] Gross, T. and Prinz, W. Modelling Shared Contexts in Cooperative Environments: Concept, Implementation, and Evaluation. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* 13, 3-4 (Aug. 2004). pp. 283-303.
- [GT96] Gross, T. and R. Traunmüller, Methodological Considerations on the Design of Computer-Supported Cooperative Work. *Cybernetics and Systems: An International Journal*, 1996. 27(3): p. 279-303.
- [Gr88] Grudin, J., Perils and pitfalls. *Byte*, 1988. 13(12).
- [Gr89] Grudin, J., Why groupware applications fail: Problems in design and evaluation. *Office: Technology and People*, 1989. 4(3): p. 245-264.
- [Ko03] Koch, M. Community Support in Universities - The Drehscheibe Project. in *Proc. Intl. Conf. on Communities & Technologies*. 2003. Amsterdam, The Netherlands: Kluwer.
- [Ko05] Koch, M. Supporting Community Awareness with Public Shared Displays. in *Proc. Bled eCommerce Conference*. 2005. Bled, Slovenia.
- [KGH02] Koch, M., G. Groh, and C. Hillebrand. Mobile Communities - Extending Online Communities into the Real World. in *Proc. Americas Conf. on Information Systems (AMCIS)*. 2002. Dallas, TX.
- [Mi01] Misch, A. Context-oriented communication support in a collaborative learning environment - Kolumbus is born. in *Proc. 24th Information Systems Research Seminar in Scandinavia Vol. 2*. 2001.
- [Wi91] Wilson, P., *Computer Supported Cooperative Work: An Introduction*. 1991, Intellect Books, Oxford, UK.