2008

CSCW and Enterprise 2.0 - Towards an Integrated Perspective

Michael Koch
Bundeswehr University Munich, michael.koch@unibw.de

21th Bled eConference
eCollaboration:
Overcoming Boundaries through Multi-Channel Interaction
June 15 - 18, 2008; Bled, Slovenia

CSCW and Enterprise 2.0 - towards an Integrated Perspective

Michael Koch

Bundeswehr University Munich, Germany
michael.koch@unibw.de

Abstract

In CSCW we are researching support for collaboration in work groups for several decades now. Web 2.0 and Social Software entered this field from another starting point recently, and quickly expanding towards support for collaboration in enterprises (Enterprise 2.0). However, the interaction between both fields is minimal. In this paper I am trying to contribute to bridging the gap by identifying the core contributions of the two fields, and how they can be integrated or used to the benefit of both fields.

Keywords: CSCW, Groupware, Social Software, Web 2.0, Enterprise 2.0

Introduction

The media have found a new hot subject: the Web 2.0. There is no large newspaper or magazine that has not yet reported about participation and collaboration of users in the Web, or about specific application classes or applications such as Wikipedia, Delicious, Google Calendar or YouTube.

Sometimes, when reading about Web 2.0 or Social Software one can receive the impression that this is a new development. This is both true and false.

While the term "Social Software" has only become relatively popular within the last two or more years, the core ideas of Social Software and especially of its application for supporting collaboration in organizations enjoy a much longer history, running back to Vannevar Bush's ideas about the Memex in 1945, and travelling through terms such as Groupware, Group Support Systems and Computer-Supported Collaborative Work (CSCW) in the 1970s to 1990s (see Allen 2004 for a good overview of this history). Most of what currently is advertised as a revolution on the Web has been there as CSCW applications years (or even decades) ago – however, not as nice and not as usable as today in the Web 2.0 with Social Software.

In my opinion the Social Software field could benefit a lot from applying CSCW research results, and vice versa the CSCW field could benefit from taking a closer look into what is happening in the Social Software field. This is especially true since Social Software is currently extending its application area to supporting teams in companies (Enterprise 2.0). However, it appears that the interaction between both fields is minimal. In this paper I am trying to contribute to bridging the

gap by identifying the core contributions of the two fields, and how they can be integrated or used to the benefit of both fields.

The Beginning

The origins of Computer-Supported Collaborative Work (CSCW) as a research field can be traced back to the early 1980s – more precisely to a workshop organized by Irene Greif and Paul Cashman. This workshop took place 1984 in Endicott House, Massachusetts, and brought together participants from different disciplines to exchange ideas and results about using IT for supporting collaborative work.

Since then, several technologies and tools have been developed and tested, and insights into understanding the process of collaboration and the process of introducing tools for supporting communication and collaboration have been obtained. For example coexistence and awareness have been worked out as the primary prerequisite for successful cooperative work (Dourish & Bellotti 1992; Koch & Gross 2006), and coordination has been operationalized (Malone & Crowston 1992).

One interesting model developed in CSCW for understanding different modes of communication in groups for example is the people/artifact framework which addresses the functional relationship between members and the tools to support collaboration, and maps out these relationships in a way that allows designers to follow the flow of information within the system (Dix et al 1993, p. 465). Figure 1 shows the core of this framework. The directional and bi-directional arcs indicate channels of communication either between participants or between a participant and the artifact.

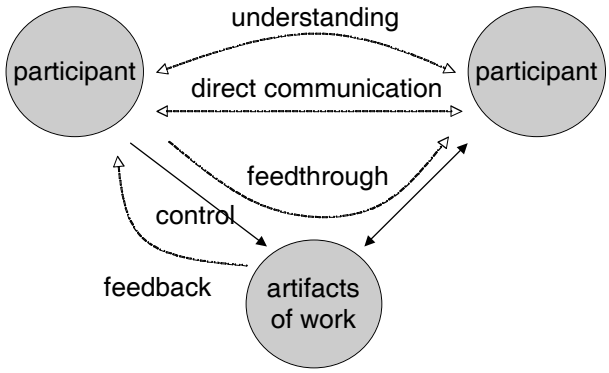

Figure 1: *People/Artifact Framework*

Computer systems for supporting collaboration are often labelled as Groupware. Marca and Bock (1992, p. 60) state that the development of Groupware was not merely another evolutionary step in the history of computer science, but *"a conceptual shift; a shift in our understanding. The traditional computing paradigm sees the computer as a tool for manipulating and exchanging data. The Groupware paradigm, on the other hand, views the computer as a shared space in which people collaborate; a clear shift in the relationship between people and information*."

This conceptual shift has far-reaching consequences. Indeed, the statement implies that Groupware is not characterized by single isolated applications, which have some collaborative aspects, nor should the computer only be seen as a means of information processing. Instead, computers are a medium for communication and collaboration. These important aspects of computers already were anticipated by visionaries like Vannevar Bush, Douglas Engelbart and Joseph Carl Robnett Licklider decades ago (Busch 1945; Engelbart 1963; Licklider & Taylor 1968).

CSCW and the Work System

Collaboration support is not only about providing technologies and tools, but about shaping socio-technical systems. The term "socio-technical system" has been coined in the 1950ies by Trist and Bamforth (1951) at Tavistock Institute London in the context of a number of studies of work organization in the British coal mining and textile industries (also see Emery & Trist 1960). In these studies the researchers found very different results emerging from the introduction of identical technology into different groups (social systems). The central lesson from analyzing the observations was that the technical system and the social system have to be co-optimized for the whole system to be successful. If a technical system is created or introduced at the expense of a social system, the obtained results will be sub-optimal. While rooted in classical workplace studies the concept was later also adapted to the usage of computer based information systems to support social groups (Mumford 1987) and to knowledge management (Coakes 2002). Here the technical system is the IT-system introduced to help the people to communicate and to collaborate, the social system is constituted of the organization and the relationships between the group members, including social actions and interactions, formal and informal expectations, roles, policies, values and group dynamics.

In workplace psychology the term "work system" is used for a socio-technical system representing clearly identifiable and separable subsystems in an organization or a company. Work systems are systems in the sense of system theory, i.e. they transform input to output. According to models from workplace psychology they can be seen as consisting of the following parts (see for example Sydow 1985):

- people (with qualifications, interests, and requirements)
- technology (machines, IT-systems, work resources, special conditions)
- organization and structure (work processes, decision making structures, communication structures)
- the primary task of the work system

The primary task/goal is of core importance for the socio-technical system, because it provides a source for motivation and for holding the system together.
CSCW research has adapted these insights to shaping systems for collaboration support. In summary, the main messages from the socio-technical systems discussion for CSCW are

- technical systems (CSCW support technology) are highly embedded in social systems
- the social and the technical subsystems should be optimized (designed) in parallel, because they influence each other
- the goal/task of the overall system should not be forgotten – it usually is a main source for the coherence of the system

The socio-technical system approach highlights the alternating dependency between social systems and technical components. Social processes are the basis for the development of technology and vice versa the technology structures the possibilities for social exchange. Giving equal weight to social and technical issues when designing new work systems is of core importance for success (Mumford 2000).

So, 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. Bowers and Benford (1991, p. 5) probably have the most general sight. They state 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*".

Other researchers emphasize the aspect of group work or group activity in CSCW. For instance, Greif (1988, p. xi) defines CSCW as "*computer-assisted coordinated activity such as communication and problem solving carried out by a group of collaborating individuals*". And for Wilson (1991) 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".

Groupware

CSCW is about understanding collaboration and about shaping socio-technical systems for supporting this collaboration. Nevertheless, there are some technologies and tools that facilitate shaping these socio-technical systems.

For these technologies and tools the term "Groupware" is used. In contrast to traditional computer systems that are primarily designed for a single user, the major goal of Groupware is to assist a group of users in communicating, in collaborating, and in coordinating their activities (Ellis et al. 1991). Ellis, Jacobson und Horvitz state that the term Groupware stands for "*computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment*" (Ellis et al. 1991, p. 40).

Groupware can include software, hardware, services or group process support (see Figure 2 for an overview of Groupware classes structured by the main mode of interaction supported, see Sauter et al., 1994). However, the positions of researchers diverge in their view of when distributed systems end and Groupware begins, i.e. what distinguishes Groupware from non-Groupware.

In our opinion the core characteristic of Groupware is the explicit attempt to reduce the isolation of users from each other (Koch & Gross 2006). 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. Lynch et al. pointed out in 1990 that "*Groupware is distinguished from normal software by the basic assumption it makes: Groupware makes the user aware that he is part of a group, while most other software seeks to hide and protect users from each other (…) Groupware (…) is software that accentuates the multiple user environment, coordinating and orchestrating things so that users can "see" each other, yet do not conflict with each other.*" (Lynch et al., 1990, p. 160)

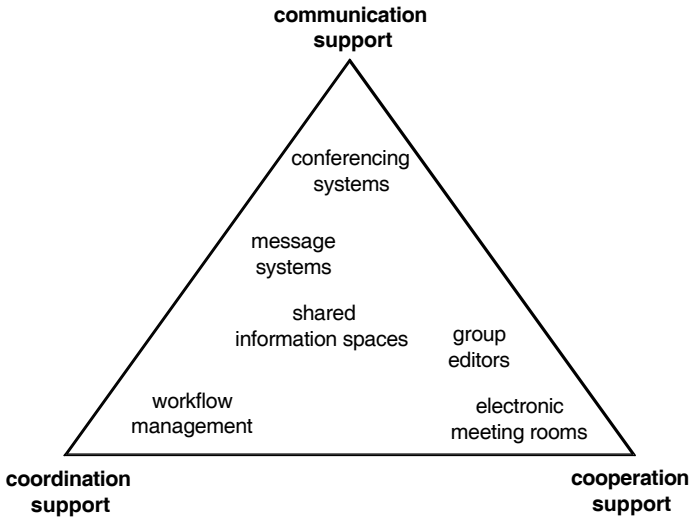

Figure 2: *Categorizing Groupware in between the three different interaction modes – communication, coordination, cooperation (Sauter et al., 1994)*

Another common characteristic that has been identified for Groupware is adaptability. Since every group is different Groupware not only has to match the needs of the group but also has to match the needs of the different group members to achieve a critical mass of users. Thus, there usually is no one-fits-all solution. Groupware either has to be very generic – medium-like -, so that users can use it in quite different ways (email is the best example for this medium aspect of Groupware), or has to be highly customizable. The customization usually should be doable for the end-user himself. A term has emerged for representing this concept: End-User-Development (Lieberman et al. 2006).

In addition to this viewpoint there is still another Groupware "definition": In technical press Groupware is often equated with Microsoft Exchange/Outlook or similar software (based on the MAPI protocol). Following this viewpoint, Groupware would only cover email, (group) calendars, (group) address books and (group) task lists.

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 (Ellis et al. 1991; Grudin 1988, 1989). 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 because – 1) many different 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 co-workers have to use the system actively (network effects, critical mass). This requires mainly a clear balance between effort and benefits for all of the users (no disparity between effort and benefits), which has to be communicated to the users, and includes the need for easy-to-use user interfaces and for a good integration.

In CSCW lots of contributions to these issues have been worked on. The most important ones are in requirements engineering and change management – introducing participatory and evolutionary development (Koch & Gross 2006).

In this context it is important to note that "implementing a system" in CSCW always means designing a complete socio-technical system, including organizational and social aspects. It is even quite common today that for the technical component of the system no completely new system is implemented, but "just" off-the-shelf tools are selected, integrated and configured.

CSCW research provides a large body of work and experience on how to design and implement information systems for supporting collaboration in enterprises (regarding the social and organizational needs of these organizations). For watching users to derive requirements in CSCW ethnographic methods have been applied to workplace scenarios (see for example Blomberg et al. 1993; Jordan 1996). By focusing in observation and the study of people at work in their normal work setting, ethnography can uncover work practices that will not get revealed by asking people directly about their work.

For actively involving users different participatory design methods have been developed (Schuler & Namioka 1993; Muller & Kuhn 1993). Participatory design is a complementary method to ethnography in which the users and other stakeholders of the software are involved in the design from a very early stage and throughout the design and development process. One example for a participatory design method is the Sociotechnical Walkthrough (Herrmann et al. 2004, 2007). This method includes different moderated workshop settings in which the whole socio-technical system is discussed or developed with the users. For describing socio-technical systems a special modelling approach has been developed that adds special constructs for socio-technical systems to standard systems modelling approaches like UML (Loser & Herrmann 2001).

The Emergence of Web 2.0 and Social Software

For Web 2.0 the most often cited characterization comes from Tim O'Reilly in his article "What is Web 2.0" (O'Reilly 2005). O'Reilly summarizes the Web 2.0 (compared to the Web 1.0) as

1) an architecture of participation,
2) mixable data sources (data centricity), and
3) easily configurable and combinable services instead of pre-packaged software.

The most important concept is participation, which means the free cooperation of as many contributors as possible without any restraints from organizations, processes, technologies or particular platforms. This is made possible by lowering the entry barriers through providing:

- usability – by being web-based and interactive (e.g. Ajax)
- "me"-centricity – the core of Web 2.0 is that every application has to yield a benefit for the single user (a direct timesaving benefit or some kind of intrinsic motivation) – in contrast to values defined by the benefit for teams and communities

Social Software is often presented as a subset of the Web 2.0 – in the form of software or services that support, extend or derive added value from human social behaviour (Coates 2005). Here, we have the concepts from the Web 2.0 again: it is about the participation of the individual – not primarily about collaboration (like in Groupware). By the way: There was an argumentation for the computer as a social medium and for the research program "The Social Web" long before the term Social Software came up (see Hoschka 1998).

The inventor of the term "Social Software" Clay Shirky stated that he had chosen the term as he was "*looking for something that gathered together all uses of software that supported interacting groups, even if the interaction was offline*". He also argued against not having chosen the term collaborative software, instead of Social Software "*because that seems a sub-set of Groupware, leaving out other kinds of group processes such as (…) play*".

McAfee (2006) summarizes the characteristics of Social Software in the acronym SLATES (search, links, authoring, tags, extensibility, signals), which we are using in a slightly adapted version (Koch & Richter 2008):

- Being able to publish contributions or edit content as easily as possible ("Authoring")
- Being able to contribute structuring metadata by tagging ("Tags")
- Add additional content and metadata by annotation and linking ("Authoring", "Links")
- Possibility to subscribe to new content ("Signals")
- Being able to find new content ("Search", "Tags")
- Modular, service oriented and data-centric design of the applications ("Extensions")

Some core application classes are related to Social Software, e.g. wikis or blogs. In the style of the 3C model of Groupware (Figure 2) we have tried to organize the software classes in a triangle with the Social Software core concepts at the edges: information management, identity- and network management and communication (see Figure 3).

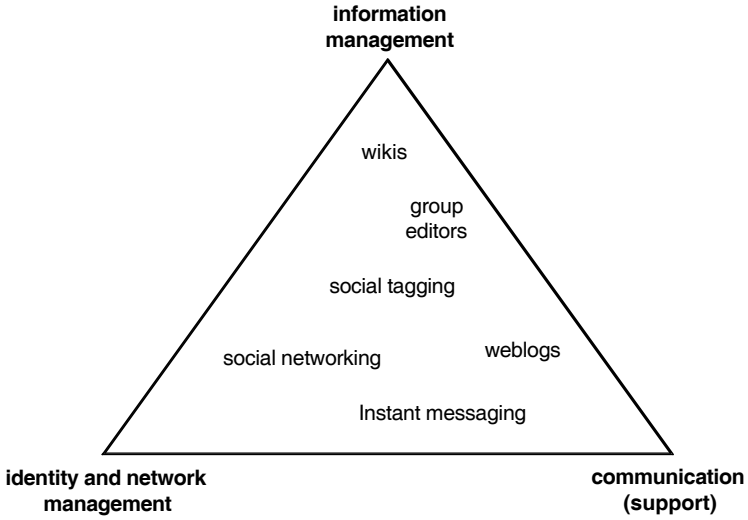

***Figure 3**: Social Software Triangle*

The Social Software Triangle gives also an overview of the possibilities of using Social Software (to support collaborative work) "*within a company or between companies and their partners or customers*" which has been called Enterprise 2.0 by McAfee (2006).

Enterprise 2.0 and CSCW

If we compare Social Software and Groupware, there truly seem to be some differences:

- Group oriented communication (in Groupware) vs. person/self oriented communication (in Social Software) – "we"-centricity vs. "me"-centricity
- Top down implementation and enforced participation (in Groupware) vs. bottom up implementation and voluntary participation (in Social Software)
- Pre-planned ways of working together (in Groupware) vs. co-evolved conventions (in Social Software)
- Small number of users over a limited period of time (in Groupware) vs. large number of users with no project limitations (in Social Software)

But looking closer, new types of Groupware no longer focus on projects and small numbers of users only, and from the very beginning CSCW stressed the importance of giving the users freedom, of focusing on user motivation, and of co-developing the solutions with the users.

Hence, since the commonalities seem to outweigh the differences, experiences from CSCW can contribute to addressing the most important difference between Enterprise 2.0 and Social Software in the Internet: the rooting of the Social Software tools in the enterprise with its inherent organizational patterns. This is important because there will always be existing structures and platforms which have to be taken into consideration when introducing Social Software in the enterprise.

Here is how we recommend addressing this issue in an integrated perspective for Enterprise 2.0:

1) Stay as close to the optimum "me"-centricity and bottom up introduction of Social Software as possible.
2) Learn from CSCW to make the introduction of collaboration tools as participatory and smooth as possible.

Firstly, as it is often mentioned in discussions about Enterprise 2.0, for successfully implementing the benefits of Social Software often the enterprise culture has to change so as to have as little hindering structures and hierarchies as possible. But not all structures in the enterprise can and should be dropped – otherwise it would no longer be an enterprise. An enterprise is more than a collection of individuals. To achieve the benefits of teams and enterprises the employees have to coordinate with each other and to adapt to common ways of doing things.

We can come close to the ideal by involving the employees as much as possible in the design of the workplace – to do a participatory design of the work systems (the whole socio-technical system including technology but also organizational and social aspects). Following the lessons learned from CSCW this participatory design and introduction should be done in an evolutionary, iterative way. So, when trying to introduce a wiki in a team for example, the team leader should involve all team members from the beginning. Ideally, one would start with a discussion in one of the regular team meetings to identify the problems to be solved with the wiki. This includes identifying the processes and other systems that are involved. Sometimes this step already helps to identify a simple, non-technical solution. If there is still room for a technical solution one then should try to get an agreement of all team members on 1) what tool should be used and what processes have to be changed, 2) what everybody has to do to make this work, 3) what everybody can expect from the approach. Finally, the solution should be reviewed regularly and be adapted (or dropped) if needed.

The need to include "some bottom-up" initiative in the introduction of Groupware can also be explained by the status of Groupware in the typical IT organization. Groupware lies in between individual productivity software – where individual value and motivation is of core importance - and large systems where strong backing by upper management is critical to success. Similarities to large systems include the need to tailor to individual settings and the likelihood of mixed reactions by group manners. On the other hand, the apparently low cost of Groupware gives it less visibility in a large organization. Grudin and Palen (1995) examined this issue on collaborative meeting schedulers and identified these bottom-up adoption patterns.

Another core lessons from CSCW is to look into a balance between effort and benefit for successful adoption. While ideally, there is a positive benefit for all users in Social Software, there often is not if it has to be implemented in a particular organizational setting. Thus, ways to balance the benefit are needed – and motivation is one of them. Basic research on social systems tells us about the difference between intrinsic and extrinsic motivation and about effects where extrinsic motivation undermines intrinsic motivation (Frey & Osterloh 2002). Extrinsic motivation is gained from exogenous salaries, intrinsic motivation from internal factors such as job satisfaction and identification with common values. The "crowding-out effect" we just mentioned says that paying users to cooperate (or linking payment to participation on Social Software platforms) can be very contraproductive by lowering or even replacing natural intrinsic motivation like the joy to work on a project or the joy in being recognized as an expert. The design of motivation systems should strengthen intrinsic motivation, e.g. by giving people more freedom, or by making people more visible (for making appreciation possible).

Having Fun with Sharepoint?

At a workshop on "CSCW and the Web 2.0" at the European Conference on Computer-Supported Cooperative Work 2007 in Limerick a participant coined the quote: "*I could use Facebook to do something seriously – but I cannot use Sharepoint to do something fun.*"

In my opinion this statement with Facebook as an example for Social Software and Microsoft Sharepoint as an example for Groupware nicely points to some differences between classical CSCW tools and Social Software that could serve as a basis for further improvement on both sides.

First the quote shows one reason why the motivation to use Social Software is higher than using classical groupware, the possibility to have fun. This may go down to the issue that Social Software is about contribution, addresses the needs of the individual user, and builds on shared

personal benefits. Collaboration happens rather "by accident". Social Software usually is built with fun and usability in mind, while Groupware long has been built with processes and algorithms in mind only.

But there is an even more fundamental message in the quote: Social Software is built for being a 'medium', i.e. being flexible to be used in different settings for different – not yet envisioned – tasks, while CSCW tools are too much focused on the task, leaving users too little freedom.

Sharepoint can only be used (easily) for one particular thing – coordinating access to documents and exchanging information in teams in forethought ways. However, Facebook (or any combination of Social Software like blogs or wikis) provides a quite generic communication platform that can be used for the tasks Sharepoint can, but also for others. Since every cooperation scenario is different, and even in one setup the needs are constantly changing, users of rigid platforms have to work around the platform more and more, while users of media-like platforms just use the medium in new ways.

But what about the efficiency of Groupware use? How to ensure that the co-workers still collaborate efficiently if we provide an open collaboration media? At least not by restricting the possibilities, but rather by helping the users to explore the possibilities and to avoid errors others have made before. For example by providing benefit oriented documentation, and by allowing the users to exchange best practices of the medium usage.

One lesson learned in Social Software that could be picked up by CSCW is "the art of letting go". Not every activity in a CSCW tool has to be controlled or lead to a particular and verifiable goal. It only has to be made sure that the activities as a whole contribute to the benefit of the organization. Anderson (2006) nicely summarizes this in a quote speaking about systems evaluation: "*to evaluate systems you have to think probabilistically – not every activity (e.g. every tag) is important/has to be perfect, but altogether it works (rule of big numbers)*". Quality is not reached through restrictions and rules but through transparency and participation. Key to success is to strike the right balance between providing the right guidelines and leaving enough freedom, i.e. "let the user decide", and monitor the evolving process. Insufficient guidance might lead to an ineffective diversity, but too much guidance might kill the important "fun factor". Consequently, the guidelines should be developed in a participative and iterative approach in close cooperation with the actual users.

Another big difficulty for the application of Social Software (in the enterprise), that could be addressed by guidelines, is the variety of media (media choice problem), which can be justified e.g. by the media synchronicity theory (Dennis & Valacich 1999). The challenge is to arrange the mutual laps and transitions of all media as efficient as possible given the enterprise-specific context.

The contributions of both fields could be summarized in the following way: Web 2.0 provides new technologies, a focus on usefulness and on the medium aspect of support, while CSCW provides insights into groups and the needs of organizations and management. Wolfgang Prinz nicely summarized this during the already mentioned workshop at ECSCW 2007 by labeling "Web 2.0 as living lab of freedom" vs. "CSCW as living lab of policies".

For successful implementation in enterprises Social Software has to be integrated in actual business practices, and to be used – or rather "lived" – by all involved parties. So, Social Software should also be adapted to the wishes of the managers who would like to keep control about the working processes. Managers are looking for dashboards (and cannot find them in Social Software yet).

Conclusion

Enterprise 2.0, the usage of Social Software in the enterprise, is a big chance. A chance, for companies as well as for researchers, to learn from successful cooperation and collaboration patterns shown by Web 2.0 platforms on the Internet.

In this paper I have tried to put Enterprise 2.0, in a broader context, and set it in context with a field that has much in common with it: Computer-Supported Collaborative Work (CSCW). Since Enterprise 2.0 is "a little bit top-down" and not purely "bottom-up" as is Social Software the insights gained in CSCW can help making Enterprise 2.0 even more successful.

The Web 2.0 is about the development and usage of new ways of communication and tools for this in the public; CSCW is about organizationally focused support of collaboration in the enterprise. CSCW should incorporate successful Web 2.0 patterns in addition to the existing Groupware and focus on making the ideas and tools from the two fields work together. Research questions can be found in learning about success factors (of Social Software) and non-success factors (of CSCW), in making the integration in business practices work, in identifying functionalities of CSCW systems that should be incorporated in Social Software systems, to support the users to choose the right media, etc.

These insights are mainly on how groups and organizations work and how to introduce technical systems (shape socio-technical systems) to support this work. CSCW can provide the theory foundation (the nature of collaboration, modes and models of collaboration, experience for collaborative system design etc.) for Enterprise 2.0. The experiences from CSCW are especially important when there is a need to consider organizational and social aspects of (a little bit top-down) setting up support for communication and cooperation in enterprises.

Thus, it is important to observe critically the evolution of the www (or Web 2.0), to implement and adapt new promising elements to support collaboration, but in the end not to forget core lessons learned of CSCW.

References

Allen, C. (2004): Tracing the Evolution of Social Software. Life with Alacrity Blog. http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html

Anderson C. (2006): The long Tail, Hyperion Publishers.

Blomberg, J.; Giacomi, J.; Mosher, A.; Swenton-Wall, P. (1993): Ethnographic field methods and their relation to design. In: Schuler, D.; Namioka, A. (eds.), Participatory Design: Perspectives on System Design, pp. 123-154, Hillsdale, NJ: Lawrence Erlbaum Associates.

Bowers, J.; Benford, S. D. (eds.) (1991): Studies in Computer-Supported Cooperative Work: Theory, Practice, and Design - Human Factors in Information Technology, Amsterdam: Elsevier Science.

Busch, V. (1945): As We May Think. The Atlantic Monthly, 176(1), pp. 101-108.

Coakes, E. (2002): Knowledge Management: A sociotechnical Perspective. In: E. Coakes, D. Willis, D., & S. Clarke (Eds), Knowledge Management in the Sociotechnical World. The Graffiti Continues (pp.4-14). London: Springer.

Coates, T. (2005): An addendum to a definition of Social Software. http://www.plasticbag.org/archives/2005/01/an_addendum_to_a_definition_of_social_software Last access 11.05.2007.

Dennis, A.K.; Valacich, J. S. (1999): Rethinking Media Richness: Towards a Theory of Media Synchronicity. In: Sprague, R.H. (Hrsg.): Proceedings of the 32nd Hawaii International Conferene on System Sciences (HICSS 32). Los Alamitos [CD-ROM, IEEE Computer Society].

Dix, A. J.; Finley, J.; Abowd, G. D.; Beale, R. (1993): Human-Computer Interaction. New York: Prenice Hall.

Dourish, P.; Bellotti, V. (1992): Awareness and Coordination in Shared Workspaces. Proc. Intl. Conf. on Computer-Supported Cooperative Work, pp. 107-114.

Ellis, C. A.; Gibbs, S. J.; Rein, G. L. (1991): Groupware - Some Issues and Experiences. Communications of the ACM, 34(1). pp. 38-58.

Emery, F. E.; Trist, E. L. (1960): Socio-Technical Systems. In: Churchman, C. W.: Management Sciences, Models and Techniques, London: Pergamon.

Engelbart, D. C. (1963): A Conceptual Framework for the Augmentation of Man's Intellect. In: Information Handling, Volume 1, Washington, DC: Spartan Books.

Frey, B. S.; Osterloh, M. (2002): Motivation - A Dual-Edged Factor of Production. In: Frey, B. S.; Osterloh, M.: Successful Management by Motivation. Berlin: Springer, pp. 3-26.

Greif, I. (1988): Computer-supported cooperative work: A book of readings. Los Altos: Morgan Kaufmann.

Grudin, J. (1988): Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. In: Proc. Conf. on Computer-Supported Cooperative Work (CSCW), ACM Press, pp. 85-93.

Grudin, J. (1989): Why Groupware applications fail: Problems in design and evaluation. Office: Technology and People, 4(3), S. 245-264.

Grudin, J.; Palen, L. (1995): Why Groupware Succeeds: Discretion or Mandate? In: Proc. Europ. Conf. on Computer-Supported Cooperative Work (Stockholm, Sweden), Kluwer, pp. 85-93.

Herrmann, T.; Kunau, G.; Loser, K.-U.; Menold, N. (2004): Socio-Technical Walkthrough: Designing Technology along Work Processes. In: Proc. Participatory Design Conferece (PDC), pp. 132-141.

Herrmann, T.; Loser, K.-U.; Jahnke, I. (2007): Socio-technical Walkthrough (STWT): a means for Knowledge Integration. In: International Journal of Learning Organisation. Special Issue: Solving Problems in Knowledge Sharing with Sociotechnical Approaches. E Coakes; E.; Ramirez, A. (eds.).

Hoschka, P. (1998): Der Computer als soziales Medium. In: GMD-Spiegel 3 / 4 (28), pp. 16-19.

Jordan, B. (1996): Ethnographic workplace studies and CSCW. In: Shapiro, D.; Tauber, M.; Traunmüller, R. (eds.), The Design of Computer Supported Cooperative Work and Groupware Systems, pp. 17-42, Amsterdam: Elsevier Science.

Koch, M.; Gross, T. (2006): Computer-Supported Cooperative Work - Concepts and Trends. In: Proc. Conf. of the Association Information And Management (AIM), Lecture Notes in Informatics (LNI) P-92, Bonn: Koellen Verlag.

Koch, M.; Richter, A. (2008): Enterprise 2.0 - Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen. Oldenburg Verlag, München.

Licklider, J. C. R.; Taylor, R. (1968): The Computer as a Communication Device. Science and Technology 76, pp. 21-31.

Lieberman, H.; Paterno, F.; Wulf, V. (eds.) (2006): End-User Development. Dordrecht, The Netherlands: Springer, Human-Computer Interaction Series Vol. 9.

Loser, K.-U.; Herrmann, T. (2001): Diagrammatic Models for Participatory Design of Socio-Technical Systems. In: Bjornestad, S.; Moe, R.; Morch, A.; Opdahl, A.: IRIS 24. Proceedings of the 24th Information Systems Research Seminar in Scandinavia, pp. 285-300.

Lynch, K. J.; Snyder, J. M.; Vogel, D. M.; McHenry, W. K. (1990): The Arizona Analyst Information System: Supporting Collaborative Research on International Technological Trends. In: Gibbs, S.; Verrijn-Stuart, A. A.: Multi-User Interfaces and Applications, Amsterdam: Elsevier Science, pp. 159-174

Malone, T. W.; Crowston, K. (1992): Towards an Interdisciplinary Theory of Coordination. Technical Report CSS TR#120, Cambridge: Center for Coordination Science, Sloan School of Management, Massachusets Institut of Technology.

Marca, D.; Bock, G. (eds.) (1992): Groupware: Software for Computer-Supported Cooperative Work. Los Almitos: IEEE Press.

McAfee, A. (2006): Enterprise 2.0: The Dawn of Emergent Collaboration. MITSloan Management Review 47(3): S. 21-28.

Muller, M.J.; Kuhn, S. (eds.) (1993): Special Issue on Participatory Design. Communications of the ACM, 36(6).

Mumford, E. (1987): Sociotechnical Systems Design – Evolving theory and practice. In: Bjerknes, G.; Ehn, P.; Kyng, S. (eds.): Computers and Democracy: A Scandinavian Challenge, Aldershot: Avebury, pp. 59-77.

Mumford, E. (2000): A Socio-technical Approach to Systems Design. Requirements Engineering 5, pp. 125-133.

O'Reilly, T. (2005): What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html Last access 5.01.2007.

Sauter, C.; Mühlherr, T.; Teufel, S. (1994): Sozio-kulturelle Auswirkungen von Groupware. In: Rauch, W.; Strohmeier, F.; Hiller, H.; Schlögl, C. (eds): Proc. 4 Intl. Symp. für Informationswissenschaft, Universitätsverlag Konstanz, p. 517-526.

Sydow, J. (1985): Organisationsspielraum und Büroautomation. Berlin: De Gruyter.

Trist, E.; Bamforth, K. (1951): Some social and psychological consequences of the long wall method of coal getting. Human Relations 4, pp. 3-38.

Wilson, P. (1991): Computer Supported Cooperative Work: An Introduction, Oxford, UK: Intellect Books.