

# Towards a CROSS FERTILIZATION Between Social Software and CSCW

Sandy El Helou, Denis Gillet, Chiu Man Yi  
Ecole Polytechnique Fédérale de Lausanne (EPFL)

## Introduction

From *Memex* [1], passing by *Groupware* and *CSCW*, to *Social Software*, several terms were initiated, spread and adopted to describe research techniques and collaborative applications aimed at supporting group interaction. Clay Shirky stated that he had chosen the term *Social Software* 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 **discussion, mutual advice or favors, and play**" [2]. Still, those arguments in favor of this new term didn't prevent Louise Ferguson from questioning whether *Social Software* is not just "a new label for old bottles" [3]. Moreover, and ironically enough, Bonni Nardi stated, " We have decided last time [CSCW 2004] that **CSCW is about play too**", and Joe McCarthy and Elizabeth went even further by "half-jokingly" describing *CSCW* as "**Computer Supported Cooperative Whatever**" in the informal call for participation at ECSCW 2001[4]. Consequently, those puzzling statements and extensions around the definition and the use of the terms *CSCW* and *Social Software*, leaves us with the open questions: Are there real added values and new distinguishing characteristics associated with the new term *Social Software*? More importantly, could those distinguishing characteristics be seen as a means to solve the problems and needs identified by previous and ongoing *CSCW* research? We argue that *Social Software* can be looked at, as a new "era", in field of "*CSCWhatever*", which, by taking a different perspective and adopting a bottom-up approach, has "democratized" and "popularized" the domain, and consequently, considerably contributed to overcoming the problems identified by *CSCW* research and applying the lessons learned from earlier *Groupware* applications. This position paper aims at supporting this claim by confronting the lessons learned from *CSCW* and the corresponding distinguishing key features of *Social Software*. Moreover, illustrative examples will be taken from eLogbook, a Web 2.0 collaborative Web Application developed at the EPFL and aiming at supporting groups' interaction.

## **Social Software: A new philosophy in Practice learning from CSCW lessons**

### **Lessons from CSCW:**

In this section, some CSCW applications drawbacks identified by CSCW researchers will be examined.

First of all, the problem of **low participation** and lack of **personal incentives** has been identified. To illustrate this problem, Grudin gave the example of the task of scheduling of a meeting [5]. Even though this is the task of the scheduler, for it to be successful, other group members might need to maintain an electronic calendar, though they might not see a direct individual benefit from doing so, as the only benefit that they can clearly identify is simplifying the scheduler's task. From this example, the need to create incentives and for individuals and trigger an active participation and involvement in interacting with the group by using the application "medium", is clearly expressed.

Second, Ackerman identifies the necessity of having flexible, nuanced and contextualized computational CSCW "apparatus" (such as roles and policies), simply because human behavior is **flexible, nuanced and contextualized** [6]. In his position paper "Applying reflection to CSCW design" [7], Dourish reminds the readers of the importance for collaborative systems to provide flexibility, the latter being considered a critical usability factor. He describes in particular, the need to support group dynamic flexibility, which he defines as the need to respond to the evolution in groups' behavior, nature and composition (e.g. membership, distribution of roles). Following this idea, he identifies the problems of the traditional CSCW systems that might force the group to adapt its behavior to the tool, because the inverse cannot be achieved. As a matter of fact, dynamic reconfiguration of those systems to take into account the groups changes is not possible for several reasons such as the fact that they had internalized or embedded the notion of "group processes", focused on very particular tasks and ignored the dynamic changes in roles assignments over time.

Last but not least, the importance of **Awareness** in collaborative spaces has been strongly stressed on in previous and ongoing CSCW work. Dourish and Bellotti [8] define awareness as "an understanding of the activities of others, which provides a context for one's own activity". Moreover, as some researches were interested in classifying awareness, others addressed the cost of interruption due to excessive notifications. In short, a need for awareness as a crucial requirement for successful group collaboration has been identified but what should be provided are provide context-dependent, relevant and personalized awareness services.

### **Social Software: A new Democratic User-Centric Bottom-Up Approach**

Social software can be thought of as the "democratization" and folks' appropriation of collaborative software applications. This approach will be discussed in details, as we describe how Social Software have successfully applied the lessons of their progenitors (CSCW), through the introduction of innovative distinctive features.

To start with, Social Software adopts a **bottom-up** approach, which consists of a solution, par excellence, to foster **active participation** and collaboration incentives.

As a matter of fact, **knowledge creation** and **collaborative authoring** have been strongly promoted and facilitated by the wave of open-source Web 2.0 applications like wikis (e.g. Wikipedia) and blogs. Similarly, **knowledge discovery and sharing** have also been facilitated by this wave of applications such as wikis, blogs, blogs search engines (e.g. Technocrati), social bookmarks managers such as (*del.icio.us*), RSS aggregators and Flickr. As a matter of fact, those Web applications foster participation because they are **User-Centered** in many ways.

First of all, they have a “**low entry cost**”; they consist of open source software easy accessible with simple URL addressing links and they are easy to use, as they don’t incorporate complicated features.

Second, interfaces are becoming much more user-friendly especially with the **Web 2.0 AJAX** techniques which increase usability and provide a better user experience as the user gets things done faster and in a smoother way.

Another factor, which has increased usability and fostered active participation, lies in the new techniques of **designing and spreading technology [9]**. Social Software applications rely on the “extreme” **participatory design policy**, adopted by designers, whereby users play a major role in the design. Applications are deployed at an early stage; no quality or usability features are guaranteed and no full specifications are provided. From there, based on the user’s reported bugs, dissatisfactions, comments and feedback and based on what kind of things users did or tried to do with the application’s offered functions, designers and developers continue implementing the features that are mostly wanted by the users who, in a way or another, willingly participated in the “design”. This ensures the continuous adaptation of the Web application to the user’s needs.

Most importantly, and this is core of the democratic bottom-up User-Centered approach: nothing is predefined and imposed on the individual. Social Software applications praise the **natural building and evolution of social networks** based on individual initiatives rather than reliance on predefined **top-down rigid group structures**. This is highly comparable to the use of **folksonomies** rather than taxonomies. So, users enter to the system as individuals and not necessarily as predefined members of a rigid predefined organizational structure. From there, groups are **organically formed**, deliberately joined/abandoned and dynamically changed. To explain, let us examine the case below, which also show how the second problem identified in the previous section is handled by Social Software bottom-up approach:

According to an interview with members of Learn-Nett (an educational Community of Practice where students from different European Universities have to just collaboratively acquire knowledge on a per-project basis), the students are not willing to collaborate because they feel the “teachers are spying on them”. They have expressed the needs for a private space over which they have full control and where they can freely share thoughts with each other. This is a good example, of how a predefined structure of a collaborative tool, **inflexibly** “labels” individuals as members of only a predefined rigid structure, not taking into account that they might need **to dynamically change their social behavior, and smoothly move back and forth from one social context to another and have different levels of information sharing**. **eLogbook**, to counter-act this problem, allowed members to naturally and easily create their own spaces. When students and teachers of Learn-Nett enter to eLogbook, they are equal, each one can create his/her own workspace and invite other people to join, there is nothing predefined, and there is no real imposed hierarchy and no Absolute Administrator or God, but rather each domain has its own god(s). A very clear demo will be shown on how **eLogbook** supports people in creating, managing

dynamic groups behavior and evolution over time in terms of nature, composition and behavior.

Last but not least, Social Software give a big importance to the issue of **awareness** and notifications. Facebook allows people to tune the notifications mechanisms according to their interest and preferences. eLogbook provide different types of contextual awareness cues taking into account the different types of awareness identified in the literature ([10],[11]). It also provides personalized notifications filtered based on the user's preferences and default eLogbook filtering rules [12] via RSS feeds and/or Email. Below, is a snapshot of the eLogbook context-Aware View. This view consists of a center or focal element chosen by the user, surrounded by four regions, each of which listing related entities of a special kind. Awareness "cues" of different types are seamlessly incorporated in every area through the use of symbolic icons, colors and the manipulation of the order in which information is displayed.



Fig. 1. Example of the context-sensitive View, the focal element being an activity



Fig. 2. Example of the context-sensitive view, the focal element being an asset

## Conclusion

To sum up, CSCW and Social Software both aim at supporting groups interaction and they both need to study human and group behavior to successfully achieve their aim. This is good a reason for which they join their "efforts", and cross-fertilize. From its side, by adopting a new democratic bottom-up philosophy, and by relying on Web 2.0 techniques such as AJAX and content syndication, Social Software Web applications has successfully overcome several problems identified with early CSCW applications and have thus, achieved more acceptability and usability and a better user experience. In the same way, the ongoing CSCW research ought to acknowledge and benefit from this new wave of Web 2.0 collaborative applications and built on their success stories. When a paper on Facebook [13], is presented in CSCW 2006 Conference, then this is a clear example of how CSCW is willing to "cross-fertilize", is welcoming and maybe even evolving towards the new bottom-up "democratic" Social Software approach.

## References

- [1] V. Bush: As we may think. In *Atlantic Monthly*, July 1945. (Online version: <http://www.ps.uni-sb.de/~duchier/pub/vbush/vbush.shtml>)
- [2] C. Allen: Tracing the Evolution of Social Software. In *Life with Alacrity*, October 13, 2004. (Online version: [http://www.lifewithalacrity.com/2004/10/tracing\\_the\\_evo.html](http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html))
- [3] L. Ferguson et al.: Selling Social Software. In New Media Knowledge, Policy Studies Institute, London, December 3, 2003. (Presentation online .pdf version: <http://www.louiseferguson.com/pdfs/NMK-socialsoftware-3Dec2003-LF.pdf>)
- [4] J. McCarthy: Computer Supported Cooperative Whatever: *Reflections on Work, Play and Passion at CSCW 2006* (Online version: [http://gumption.typepad.com/blog/2006/11/computer\\_support.html](http://gumption.typepad.com/blog/2006/11/computer_support.html))
- [5] J. Grudin: Why CSCW applications fail: In *Proceedings of the 1988 ACM conference on CSCW*, Portland, Oregon, United States, 1988, pp. 85 – 93.
- [6] M. Ackerman: The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. In *John Carroll (ed.), HCI in the New Millennium*, Addison-Wesley, 2001.
- [7] P. Dourish; V. Bellotti: Awareness and coordination in shared workspaces. In *Poc. Computer supported cooperative Work (CSCW'92)*, 1992.
- [8] P. Dourish. Applying Reflection to CSCW Design. Position paper for Workshop "*Reflection and Metalevel Architectures*", European Conf. Object-Oriented Programming, Utrecht, Netherlands, 1992. (Online version: <http://www.laputan.org/pub/utrecht/dourish.text>).
- [9] D. Boyd. The Significance of Social Software. *BlogTalks Reloaded: Social Software Research & Cases* (ed. Thomas N. Burg and Jan Schmidt), Norderstedt, 2007. pp 15-30.
- [10] C. Gutwin et al.: Workspace Awareness for groupware. In *CHI Conference Companion*, 1996, pp- 208-209.
- [11] Nutter D., Boldyreff C.: Historical Awareness Support and Its Evaluation in Collaborative Software Engineering. In *Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003, pp.171.
- [12] S. El Helou; D. Gillet; Chr. Salzmann ; Y. Rezik: Feed-Oriented Awareness for eLogbook Mobile Users. In *Interactive Mobile and Computer Aided Learning*, Amman Jordan, April 2007. (selected to be printed in IJMLO, International Journal on Mobile Learning Organization.
- [13] A face (book) in the crowd: social Searching vs. social browsing. In *Proceedings of the 2006 20th anniversary conference on CSCW*, Banff, Alberta, Canada, 2006, pp. 167-170.