Turning Web 2.0 Social Software into Versatile Collaborative Learning Solutions

Denis Gillet, Senior *Member, IEEE,* Sandy El Helou, Chiu Man Yu, and Christophe Salzmann

***Abstract*—In the framework of the European Integrated Project PALETTE, the École Polytechnique Fédérale de Lausanne (EPFL) is developing the eLogbook Web 2.0 social software. The purpose of eLogbook is to support tacit and explicit knowledge management in communities of practice. It can be customized by the users to serve as an asset management system, as a task management system or as a discussion platform. In this paper, the innovative Computer-Human Interaction features of eLogbook are introduced and its deployment scenario to support collaborative laboratory activities in engineering education is described. The main idea is to sustain interaction for learning purpose within self-organized teams that integrate -on a seamless level- both human actors (students, teaching assistants) and non-human actors such as laboratory equipments or software agents.**

***Index Terms*—Social Software, Engineering Education, Remote Laboratory, Computer-Supported Collaborative Learning**

I. Introduction

In the framework of the European Integrated Project PALETTE the École Polytechnique Fédérale de Lausanne (EPFL) is developing the eLogbook Web 2.0 social software. The purpose of eLogbook is to support tacit and explicit knowledge management in communities of practice (CoPs). It can be customized by the users to serve as an asset management system, as a task management system or as a discussion platform.

From an ethnographic point of view, the *Social Software* name encompasses nowadays the idea of computer-mediated interaction. In addition, the *Social* adjective, especially in the Web 2.0 context [1], underlines interactions related neither to professional nor to educational activities. In other words, social software mostly supports informal interaction between peers for networking and sharing purposes. The impact of such interaction within the Internet generation (the people born after 1991) is tremendous, not only for keeping them busy, but also for shaping their mind, developing their awareness, enriching their knowledge, and conditioning their behaviors. It is in fact a powerful paradigm for learning, without noticing, the skills that the knowledge society is looking for and that the traditional education system has been largely unable to develop. Hence, we are facing an increasing impact of social learning as defined by Bandura [2], even if not fully recognized, in the educational landscape. This generation has fully integrated the technology without explicitly studying it. Young people form and operate in distributed virtual teams without having had prior practice at school. Finally they instinctively use the Internet as their preferred library.

From the previous remarks, it is clear that the idea of introducing social software in traditional education is attractive for allowing students to acquire knowledge, through a sort of subliminal, effective and smooth learning process while willingly taking part of enjoyable interactive situations mediated by interesting/motivational artifacts. Such an approach may not be applicable in an engineering school to study how to derive the Maxwell equations, but it has however a huge potential for spreading and supporting project-based learning in higher education. More specifically in the engineering context considered in this paper, it could effectively support collaborative laboratory activities.

The above statement raises the question of acceptability (added-value, usefulness) and appropriation of social software solutions and social learning paradigms as substitutes for the somewhat successful *Computer-Supported Collaborative Work (CSCW)* or *Learning (CSCL)* approaches. These questions are elaborated in sections 2 and 3 as a motivation for the definition of the eLogbook Computer-Human Interaction features introduced in Section 4.

Then, Section 5 describes the eLogbook deployment as a general-purpose collaborative workspace for project-based learning activities at the EPFL. The current organization of such collaborative activities [3], despite is success [4], has not fully exploited all its potential for really developing professional project management skills. Moreover, for handling the complexity of such activities, educators have to rigorously plan activities and modalities. This paper presents in Section 6 a tentative to invert this trend by introducing *Social Software*.

This work was supported in part by the Swiss Federal Institutes of Technology in Lausanne (EPFL) and by the European Union in its sixth framework program (*Palette* Integrated Project).

Denis Gillet is with the Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, CH-1015 Switzerland (phone: +41 21 693-5168; fax: + 41 21 693-2574; e-mail: denis.gillet@epfl.ch).

Christophe Salzmann, Chiu Man Yu, and Sandy El Helou are with the Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, CH-1015 Switzerland (e-mail: firstname.lastname@epfl.ch).

II. Social Software

A. CSCW Perspective

It is worth to consider *Social Software* from a *Computer-Supported Collaborative Work (CSCW)* perspective before discussing its introduction in an educational setting.

From *Groupware* and *CSCW* to *Social Software*, several terms were initiated, spread and adopted to describe research techniques and collaborative software that support 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" [5]. However 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" [6]. Moreover, 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 [7].

Consequently, those puzzling statements and extensions around the definition and the use of the terms *CSCW* and *Social Software*, leave open questions: Should and eventually could *Social Software* be distinguished from *CSCW* and if so where should the boundary be drawn and what could be the added values and 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 depicted by the 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. We also argue that until now, *CSCW* is still considered to be tightly associated with serious matters while *Social software* is misleadingly associated with everything else. The correction of this bias is indeed an enabler for innovative *CSCL* approaches.

B. Lessons learned from CSCW

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 scheduling a meeting [8]. How can the initiator of a meeting create incentives to trigger an answer from the invited participants through an application "medium".

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** [8]. In his position paper " Applying reflection to *CSCW* design" [9], Dourish reminds the readers of the importance for collaborative systems to provide flexibility, the latter being considered a critical usability factor. In particular, he describes 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 by not least; the importance of **Awareness** in collaborative spaces has been strongly stressed on in previous and ongoing *CSCW* work. Dourish and Belloti [9] 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.

C. New Social Software 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 adopted a **bottom-up** approach, which consisted of a solution per excellence to foster **active participation** and collaboration incentives. As a matter of fact, **knowledge creation** and **collaborative authoring** have been strongly promoted and facilitating by the wave of 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 notably wikis, blogs, blogs search engines (e.g. Technocrati), social bookmarks managers such as (*del.icio.us), RSS* aggregators, Flickr. Those Web applications foster participation as 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, their user-interfaces are friendlier especially with the **Web 2.0 AJAX** techniques that 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 [10]. Social Software applications rely on the “extreme” **participatory design policy**, adopted by designers, whereby the user plays a major role. 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 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 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.

Users enter to the system as individuals and not necessarily as members of a rigid organizational structure. From there, they deliberately choose to join or abandon a group. Consequently, there is nothing preexisting, nothing predefined, groups are **organically formed** and evolve naturally, which solves the second problem identified in the previous section.

According to an interview with members of Learn-Nett (an educational Community of Practice where students from different European Universities have to 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 traditional *CSCW* approach does not take into account that students 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.

The next section shows how, from the above analysis, typical *CSCW* applications could be deconstructed and integrated in a new interaction model suitable to promote social learning using social software or, in short, Software-Mediated Social Learning (*SMSL*). In this acronym compare to *CSCW*, replacing computer by software is an attempt to also move towards ubiquitous learning relying on devices other than computers, such as multimedia players or smart phones.

III. Interaction Model for Software-Mediated Social Learning

From a general point of view, **people** use interaction technology to socialize, work and learn together through informal or formal **activities**. For these purposes, they establish synchronous or asynchronous discussion threads using phones, chat, email, blogs or wikis. They also populate distributed or centralized repositories. In these frameworks, they exchange digital information (**artifacts** or **assets)** at any time or at specified deadlines (**deliverables**). They also keep informed regarding the presence, the emotions and the activities of others trough audio or video conferencing, as well as through embedded dynamical **awareness** cues such as emoticons.

Without trying to provide a thorough classification of all *CSCW* solutions supporting the above situations/scenarios, we propose to map most of the mediated interaction alternatives using three fundamental entities: the actors, the activities and the assets. The definitions for these three entities and their possible relationships constitute what we call the 3A Model [11]. They are detailed thereafter and illustrated in Figure 1.

Fig. 1. 3A interaction model.

An **actor** is any entity capable of initiating an event in the collaborative environment. It can be a person, a Web service, a software agent or even an online physical device. In that sense, the proposed definition of an actor is broader that the traditional definition of social software users or community members. The next section will show how this extension suits the social software requirements for supporting collaborative laboratory activities. From a constructivist perspective, an actor is an **artifact doing/producing** something, i.e. an agent or an instrument corresponding to “who?” or “which?”. Awareness related to actors can correspond to emotional, social or contextual statuses (i.e. online/offline, available/busy user or machine).

An **asset** is any kind of resource produced by or shared between community actors. The proposed definition goes beyond the typical digital assets like rich-text documents or multimedia resources. It can also include as example discussion threads or wiki pages. From a constructivist perspective, an asset is an **artifact done/produced** somehow, i.e. a product corresponding to “what?”. Awareness related to assets can correspond to preview, completion state (pending, under construction, completed, …), revision or moderation state, ownership, accessibility or historical data (trace) [12].

An **activity** is the formalization of a common objective to be achieved by actors such as discussing topics or completing tasks. From a constructivist perspective, an activity is an **artifact for doing/ for producing** something, i.e a purpose corresponding to “why?”. Awareness related to such artifact could be associated membership, context corresponding to “where?”, deadlines or deliverables corresponding to “when?”).

Events or Actions related to these three main entities are governed by Protocols corresponding to “how?”.

This model is compatible with computer-supported collaborative work solutions or social software dedicated to discussions, activity management or asset management; without privileging (putting the emphasis on) any of them with respect to the others.

IV. eLogbook Computer-Human Interaction Features

To fulfill the social software requirements stated in Section 2 and to comply with the 3A interaction model proposed in Section 3, the eLogbook social software has been designed. It relies on a model-view-controller architecture and a database to centralized relevant elements such as assets, metadata, contextual information, protocols or user preferences. Three different graphical interfaces are proposed: The context-sensitive, the content-oriented and the 2D-mapping ones. The content-oriented interface is similar to a mailbox that lists entities of a specific type (activities, actors, assets) based on users’ selection criteria. The 2D-mapping interface can be best described as a social network graph made of entities and displaying different types of relations existing between them. The default and currently the most elaborated interface is the context-sensitive one that is described below.

A. *Context sensitive interface*

The context-sensitive graphical user interface maps the 3A interaction model (Figure 2). It integrates three lateral areas corresponding to actors, activities and assets that are located on the left, top and right, respectively. In addition, a bottom area dedicated to deliverables is integrated for awareness purposes. These areas are scrollable lists (white arrows) in which elements can be added using the corresponding “+” signs.

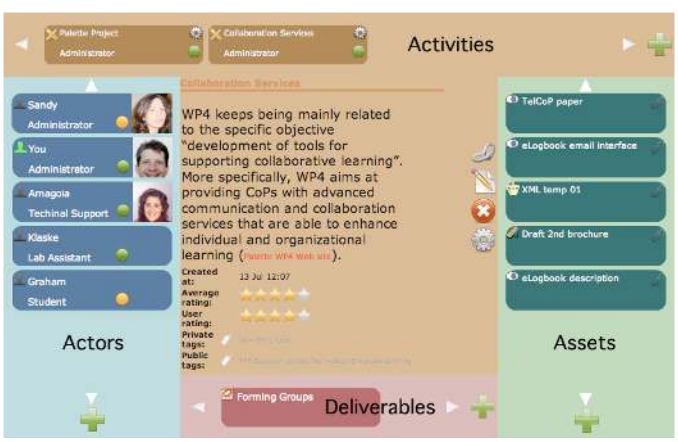

Fig. 2. eLogbook context-sensitive interface with an activity as focal element.

When an actor, an activity, an asset or a deliverable represented by a colored rectangle is selected, it is displayed as a focal element at the center. The color of the focal element corresponds to the area from which it is selected to ease the identification of its type. When the focal element is chosen, the surrounding areas are updated to display the associated actors, activities, assets and deliverables. Embedded indicators display the relationships between the focal element and the listed entities. Possible related actions that the current user is allowed to perform are also accessible through icons. Awareness “clues” of various types are seamlessly incorporated in every region through the use of symbolic icons, colors, and display orders of information. For example, deliverables with earlier deadlines are highlighted in red and appear at the beginning of the deliverable list while those with longer deadlines appear farther.

The content of the focal element includes a rich-text description that can be defined and updated thanks to an embedded wiki-like editor. By editing the description of assets and linking them sequentially, a blog can be obtained.

With eLogbook, users can naturally and easily create their own spaces as activities. They are all 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 Big Brother, everyone is in charge of his/her own domain and can share full or partial responsibility with others. In addition to the creation and the management of the core entities, tags, roles, links and rating can be defined updated an traced. They fully conditioned the eLogbook user and team experience.

B. *The online experiments as live assets*

As described in Section 3, an actor can be an online physical device. When such an actor is selected as focal element, its expended description frame provides all the necessary features to observe and interact with it. In the scenario described in section V, the online device is an electrical servo drive that can be manipulated remotely. The interface (Fig. 3) includes a scope displaying real-time measurement signals acquired on this real system. Configuration parameters like excitation or controller settings can be modified. A live augmented reality view of the drive is also provided.

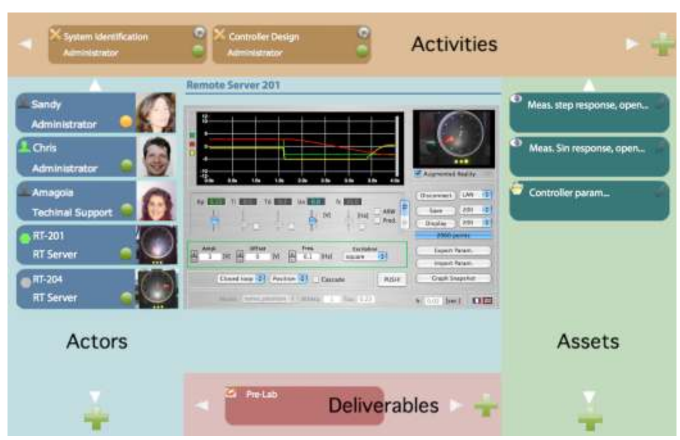

Fig. 3. eLogbook context-sensitive interface with an online device as focal element.

C. Alternative interfaces

In addition to Web-based access, eLogbook also supports information delivery through a non-intrusive email-based interface. Thanks to this interface, the users can manage their activities, assets and awareness in a ubiquitous way. This alternative lightweight interface first facilitates the appropriation of eLogbook. As a matter of fact, novice users can share knowledge artifacts and be aware of ongoing activities just by using their familiar email client software. Second, it eases eLogbook access when connecting using smart phones or PDAs. In the coming future, eLogbook will support RSS feed as another information delivery means for mobile users. The compact format of RSS feeds is particularly useful for mobile users subjected to device constraints.

Information relevance is a major concern in eLogbook. To avoid overcrowded interfaces, excessive interruptions and unnecessary alerts, it is important to dynamically adapt listed entities and notifications to the user context, e.g. device and situation. An adaptive notification filtering system for eLogbook is currently under developments.

V. eLogbook Deployement in Collaborative Laboratory Activities

A. Integration scenario

The educational framework in which eLogbook will be introduced is the automatic control laboratory sessions offered to students from electrical, micro and mechanical engineering at the EPFL. The students have five mandatory two-hours such sessions planned in their study program during their last bachelor year. So, the question is how to really take advantage of a 10-hours allocation for laboratory project activities spread during a full semester to consolidate the students' competences in automatic control and to developed engineering project management skills. Till now, the learning scenario relies on a flexible learning approach (in terms of modalities) with a strict program (in terms of task planning). The flexible learning approach relies on a free choice for the students regarding the place and time of study. During the last six years, students have been able to choose to carry out their laboratory assignments either within the laboratory premises or remotely using the same *eMersion* Collaborative Web-based experimentation solution [3]. An introduction session followed be three experimentation modules including preparatory and actual experimentation tasks for which only the sequence was imposed has been implemented.

The new envisioned scenario is to bring additional flexibility, not only in the modalities, but also in the team composition, in the task definition, and in the Web-based learning environment integration. The deployment of the Software-Mediated Social Learning model and eLogbook proposed in section 3 and 4, respectively, can effectively support such an objective. The details of the flexible implementation are given bellow at the modality, team, task and software levels.

Modality: The students will be able (as currently), to chose to carry out their activities either on campus in the laboratory premise during fixed schedule sessions or at distance (from a computer room or from home) at any time. A face-to-face introduction session will be organized at the beginning of the semester to present the objectives and the learning scenario. This session will be recorded and posted as a rich video podcast embedding slides.

Team: In the current scenario, the students are required to work by group of two for all the semester. In the new setting, they will be able to form group of two to 6 participants and be able to move from one group to the other during the semester. In the social learning model, people learn from their peers having complementary or additional competences. To mimic this model in the academic context, the idea is to ask teaching assistants to become team members with a position equivalent to the other students. The only difference will be that probably they will bring more competences in the group. As trust is essential in a social context, the teaching assistant will not be involved at all in the final evaluation of the group achievements. If a student move from one group to another during the semester, he or she will keep all the assets related to him or her.

Task: Currently the detailed task definition does not leave enough room for creativity and initiative. In the new setting, the team will have to negotiate a common objective and to submit it to the Professor in a given time frame. This objective will include a selection of one of the control methodology presented in the textbook, a proposal for its implementation on one of the available laboratory setups, and the description of the approach chosen for the validation of the results. At the end of the semester, the team will submit a joint final report. Then, each team member will present in an oral examination a part randomly drawn of the common work.

Software: Currently, the students have to learn and use the *eMersion* environment to complete collaboratively their laboratory assessments. With the reduction of the hour allocation in the study programs that occurred during the recent years for the laboratory activities, the learning phase of such a complete environment is too important and is not affordable anymore. In the new settings, the students will access a simple teleoperation widget available either as an independent Web application or as a eLogbook agent (actor) to configure their laboratory equipments and to conduct their predefined experiments locally or remotely. This widget enables to transfer measurements and settings to and from various repositories (from simple local hard drives, removable storage devices or mailboxes, to eLogbook, through share disks or alternative social software). Other online services enabling data analysis will be provided also either as Web applications or as eLogbook agents. Only the students finding the added value of the eLogbook features high enough for empowering their interaction and learning activities will choose to use it. As an incentive for such a purpose, a short tutorial will be given during the introduction session and also posted as podcast. One should mention that eLogbook is designed to support all the flexibility levels envisioned in this section. As a consequence, the probability that the majority of the participants will adopt it is quite strong. However, leaving the room for students to use alternative solutions (or nothing) is in our scenario essential for reaching a high acceptability level. As a matter of fact, the underlying principle behind social software is the freedom for people to use them or not. This freedom should be kept when introducing social software in an educational framework.

B. *Validation scenario*

Being simultaneously a trial from a Computer-Human Interaction and from an educational perspective, the proposed scenario will be carefully evaluated. Three instruments will be use for that purpose: Mining of the eLogbook database, online questionnaires and interviews. It is also envisioned to ask the students to give a critical evaluation of their selection of tools in their final report; engineers with university degrees being not only responsible to solve problems, but also to chose the right and most effective tools to do so in most companies.

VI. Conclusion

This paper motivates and presents a global strategy to introduce social software as a versatile support for collaboration learning activities. This strategy relies on the assertion that traditional *CSCW* solutions are not flexible enough to reach true adoption by a young generation of students accustomed to social software and continuously evolving in their interaction practices. The solution proposed to promote and adapt social software for collaborative learning relies on an interaction model that integrates three core entities (actors, activities and assets) and enables the design of a versatile and contextual shared space called eLogbook.

The eLogbook features implemented based on the proposed model are especially innovative from two points of view. First, its context-sensitive interface handles seamlessly the three mentioned entities enabling to use eLogbook as a collaborative activity management system, asset management system or discussion platform. Second, actors can be of either a human or a non-human nature enabling the establishing of and the interaction in an hybrid community integrating for example students and online physical equipments.

The eLogbook Web 2.0 social software is finally integrated as a solution to introduce more flexibility in collaborative laboratory activities based on a pedagogical scenario aiming at developing autonomy, responsibility and project management skills among Bachelor students enrolled in engineering curricula.

References

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

[2] A. Bandura, "Social Learning Theory," New York: General Learning Press, 1977.

[3] D. Gillet, A. V. Nguyen, and Y. Rekik, "Collaborative Web-based Experimentation in Flexible Engineering Education," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 696-704, 2005

[4] A. V. Nguyen, Y. Rekik, and D. Gillet, "Iterative Design and Evaluation of a Web-Based Experimentation Environment," In the book "User-Centered Design of Online Learning Communities," IDEA Group Inc. Publishers, pp. 286-313, 2007.

[5] 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)

[6] 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)

[7] 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_suppor.html)

[8] J. Grudin: Why CSCW applications fail: In *Proceedings of the 1988 ACM conference on CSCW*, Portland, Oregon, United States, 1988, pp. 85 – 93.

[9] P. Dourish; V. Belloti: Awareness and coordination in shared workspaces. In *Poc. Computer supported cooperative Work (CSCW'92*), 1992.

[10] 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.

[11] D. Gillet, S. El Helou, Y. Rekik, Ch. Salzmann: Context-Sensitive Awareness Services For Communities of Practice, 12th International Conference on Human-Computer Interaction (HCI2007), Beijing, 22-27 July (2007).

[12] D. Nutter, C. Boldyreff: Historical Awareness Support and Its Evaluation in Collaborative Software Engineering. Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (2003) 171.