Applying Web 2.0 Design Principles in the Design of Cooperative Applications

Niels Pinkwart

Clausthal University of Technology, Department of Informatics
Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany
niels.pinkwart@tu-clausthal.de

Abstract. “Web 2.0” is a term frequently mentioned in media - apparently, applications such as Wikipedia, Social Network Services, Online Shops with integrated recommender systems, or Sharing Services like flickr, all of which rely on user’s activities, contributions, and interactions as a central factor, are fascinating for the general public. This leads to a success of these systems that seemingly exceeds the impact of most “traditional” groupware applications that have emerged from CSCW research. This paper discusses differences and similarities between novel Web 2.0 tools and more traditional CSCW application in terms of technologies, system design and success factors. Based on this analysis, the design of the cooperative learning application LARGO is presented to illustrate how Web 2.0 success factors can be considered for the design of cooperative environments.

Keywords: Social Software, Cooperative Applications, CSCW, CSCL.

1 Introduction

“Social Software” is a term frequently mentioned in public media - apparently, the wide success – or at least recognition and usage – of “Web 2.0” applications such as Wikipedia, Social Network Services such as Facebook, Online Shops with integrated collaborative filtering based recommender systems, or Sharing Services like flickr, all of which rely on user’s activities, contributions, and interactions as a central factor, is fascinating for the general public.

Traditionally, the research field that investigates technology support for interacting and collaborating groups and, consequently, should be the “research home” for Social Software, has been CSCW. Yet, a review of recent conferences in the field of cooperative systems shows that the ties are not as strong as one would naively suspect. While there were, for instance, two Social Software related workshops at ECSCW 2007 and there was one paper session on “Social tagging and recommending” at CSCW 2006, very little has been before that. It cannot be stated that the origin of Social Software was in CSCW – in fact, the research community in the field of cooperative work rather slowly seems to make connections to the phenomenon of Social Software. But if the advent of Social Software was not a result of systematic CSCW research in that research results achieved a public recognition and were widely

Y. Luo (Ed.): CDVE 2008, LNCS 5220, pp. 229–236, 2008.

adopted, what are then the relations between the software tools investigated in the CSCW field, traditionally named groupware tools, and Social Software? This paper discusses criteria where successful Social Software applications typically differ from traditional CSCW tools, and then shows an example to illustrate how Web 2.0 success factors can be used to inform the design of small-group oriented and focused CSCW applications.

2 Groupware and Social Software

Unsurprisingly, the overall aims of groupware and Social Software systems are very similar. While still no unique formal definition for the term "groupware" exists, there is large agreement on that groupware systems are designed to support intentional group processes [1], serve groups of users with a shared aim or goal [2], and enable users to collaborate via shared media [3]. Essentially, an extensive part of groupware and CSCW research investigates the interrelations between social group processes and collaboration technology design – often for business and work applications, as the letter "W" in the term CSCW suggests.

The term "Social Software" is much younger than the term "groupware". It isn't until 2002 that this term came into more common usage after a "Social Software Summit". According to Clay Shirky, the organizer of that summit, Social Software can be seen as a characterization of all kinds of systems that support group interaction even when that interaction is offline. Clearly, this characterization is almost identical to what constitutes "groupware" according to the accepted definitions of many researchers. There is thus no sharp definitional distinction between the concepts of groupware and Social Software. When comparing typical groupware tools to prominent examples of "Social Software" however, some characteristic differences between these two kinds of systems can indeed be found. These are discussed in the following sections.

2.1 Application Areas

CSCW and groupware are traditionally oriented towards supporting group work and enabling collaborators to interact efficiently and effectively. There is a much greater heterogeneity in Social Software tools in this regard. These systems also target fields like hobbies, leisure, or play, and consequently the tools are less driven by goals like productivity and efficiency. In fact, the organizer of the Social Software summit in 2002 explicitly refused to use the terms "groupware" or "collaborative software" because these "*had become horribly polluted by enterprise groupware work"* and "*seem a sub-set of groupware, leaving out other kinds of group processes such as discussion, mutual advice or favors, and play"* [4]. However, it should be noted that there are indeed work or business related Social Software tools such as Social Networking Services for professionals like linkedin or Xing.

2.2 Control

Social Software systems are typically very open: they delegate a lot of control to the users and the user community. For example, Wikipedia entries are not centrally "reviewed" or "edited" by default – the quality control works largely based on social protocols, supported by some technology such as logging entries and keeping version histories. Many Social Software systems address the problem of delegating control to its users while at the same time offering a trustworthy environment through a reputation system. The stars for Ebay users are a very prominent example for this. While such a reputation system may still cause problems and may be abused [5], the status and rights of users in Social Software systems are different from typical groupware applications: There is usually no central authority that assigns the status based on company organization or some other hierarchy. Instead, the user activities and how these are received by others are the basis for reputation in the system.

Also, there is little "process control" in Social Software systems. This is different in many groupware tools where the system side control about possible user actions is an important factor: As the early definitions of CSCW suggest, groupware tools are often *about* scaffolding a group collaboration process – this implies a certain intervention in the options of single users in order to coordinate the overall group workflow. The development of standards such as BPMN [6] for formally describing complex collaborative and transactional business processes or IMS-LD [7] for specifying group learning designs indicates the importance of being able to precisely express a process structure in CSCW applications. Where predefined processes exist in Social Software at all, these are usually much simpler and not strictly enforced on users, such as the feedback system in Ebay.

2.3 Technology Requirements

A considerable portion of CSCW research and the groupware tools that this research produces has always been devoted to studies how state-of-art technologies can be used to support group interactions. Examples of recent work include big displays and their impact on collaboration in the medical domain [8] or the use of mobile phone technology for collaboration [9]. Most Social Software, on the other hand, is rather "low tech" client-side and requires not more than Web access and a simple piece of software, often only a browser without add-ons that would require software installations. AJAX or DHTML concepts – technically relying on Javascript and XML – build the base for a lot of Social Software systems, including last.fm, amazon, Ebay, and many more. The avoidance of expensive or proprietary technology enables many users to access the systems (cf. next point). Even today's technically most demanding Social Software tools still run on most standard home PCs with standard network connections, as exemplified by 3D virtual environments like Second Life.

2.4 Success Factors

Participation is the key success factor for Social Software, since these systems live from the (inter-)actions of their user communities. Most successful Social Software tools are therefore extremely easy to use and do not require complicated software installations and configurations. Also, the benefits of the systems are clearly visible for the users and often also available for non-members. For instance, Wikipedia and flickr are open for everybody, and also the "Social Software" aspects of amazon.com (recommendations, reviews etc.) are visible without even logging in. These two factors of usability and immediate benefit motivate the system usage and increase the total participation in the system – which, on the other hand, is a prerequisite for the success of the system: a Social Software tool without a large user base would not be called successful.

CSCW tools, on the other hand, are frequently tailored towards smaller but more structured groups and group processes. Prototypical systems like shared calendars [10], collaborative text editors [11] meeting room technologies [12] or shared workspace systems [13] do not need huge user communities. Naturally, their quality and practical value are largely determined by productivity or functionality gains – i.e., how much support the tools provide for the group work process. The measurement of system success is a research question, its investigation may involve aspects of sociology, psychology, economics, computer science, or the specific domain targeted with the groupware tool.

2.5 Algorithms

CSCW research and the corresponding groupware systems involve a wide variety of algorithms, including, for instance, methods for controlling concurrent text editing [14], algorithms for calculating and displaying awareness information [15], and many many more. For Social Software systems, the one by far most prominent and widely used type of algorithm is collaborative filtering [16]. Through their actions in the system, users get associated to system artifacts in various ways. Examples include buying or looking at books at amazon.com, entering profiles in online dating services or tagging images on flickr. In any of these cases, the system then uses this information to recommend artifacts (products or users) to other users. While the specific calculation details clearly vary between systems, the general principle of building on the large user base and using their actions to generate the knowledge and added value of the system is at the algorithmic core of Social Software. Here, collaborative recommendation algorithms play a key role.

2.6 Summary

In summary, the above comparison shows that groupware tools and Social Software applications have differences, but these differences are not overwhelming. Their common point is their aim: facilitating group interactions and communications. The most important differences are in the fields of technology requirements, degrees of user control and application areas. Given the practical success of many Web 2.0 applications, these differences are chance for CSCW researchers to learn in order to improve the level of collaboration and the practical impact of the systems they design.

3 Applying Social Software Principles to CSCW Design: Collaborative Filtering in an Argumentation System

The sequel of this paper shows an example scenario which illustrates that the listed differences between groupware and Social Software are not a necessary requirement or a strict must. In fact, the “borderline” between groupware and Social Software systems is of course not sharp, and Social Software design principles can even inform the development of CSCW tools. The presented example system LARGO is rooted in the domain of training of legal argumentation skills. The design of this system was not driven primarily by the idea of developing a full "Social Software" tool for legal argumentation. To the contrary, LARGO is designed for rather small groups where users have to work through a well-specified task without having a great degree of control. Also, as an educational technology system, the success of the system is finally subject to empirical studies of learning and not a question of widespread usage. Yet, LARGO uses collaborative filtering and recommender system technology along with user activities like markup and tagging, all characteristics of Social Software, to estimate the quality of student's solutions though actions of other users in the system. LARGO thus exemplifies how Social Software design principles can be applied and "channeled" into goal-oriented, serious cooperative applications.

The training of legal argumentation skills is central for advocates. There are only a few systems which support users in the acquisition of these kinds of skills [17, 18]. One reason for this is that legal argumentation is an ill-defined domain - for a computer, it is a very hard task to judge whether a user-provided textual argument is good or not. Even professional judges sometimes disagree on that. The LARGO system [19] is designed to teach a group of users legal argumentation skills by allowing them to analyze examples of expert argumentation. In LARGO, these examples are transcripts of US Supreme court oral arguments. A transcript analysis is done by marking up the text transcript and annotating passages with typed descriptions, which can be put in visual relation to each other, thereby forming an argument diagram.

Figure 1 contains three different argument diagram parts which mark up a textual transcript. Every entry in a diagram can be linked to a specific paragraph in the text, and may also be linked to other diagram elements (not shown in the figure). The available types for the diagram elements correspond to an argumentation model [20]. For instance, a "test" represents a decision rule proposed by an attorney, and a "hypothetical" stands for a challenging scenario, posed by a judge, to challenge a decision rule. The goal of using LARGO is to create a visual representation of the textual transcript, to reflect upon it in order to understand the often complex and implicit argument, and thereby learn the principles of argumentation.

LARGO analyzes the structure of user-created argument diagrams and gives feedback on the structural aspects of the diagram (cf. [19] for details). This feedback is intended to help users create good argument diagrams that are structurally related to the transcript markups in a reasonable way. Yet, users may have difficulties in understanding, e.g., the essence of a proposed decision rule, as evidenced by a poor paraphrase in the corresponding "test" node they add to their diagram. Obviously, this is very hard to detect by the system, since it involves interpretation of legal argument in a textual form. It is hard to tell for a human if a description is an adequate summary of the test as formulated by the attorney during the argument or not – or, put in the terminology of Social Software, if the tag really matches the content. For a computer program it is certainly not easier to do this.

The annotations together with the fact that other users are working with the system on the same task can help here. This combination enables a quality heuristic for single argument components based on collaborative information filtering. In the LARGO implementation variant of the collaborative filtering method, users are asked to rate samples of other's work. For selected important parts of the transcript, after having

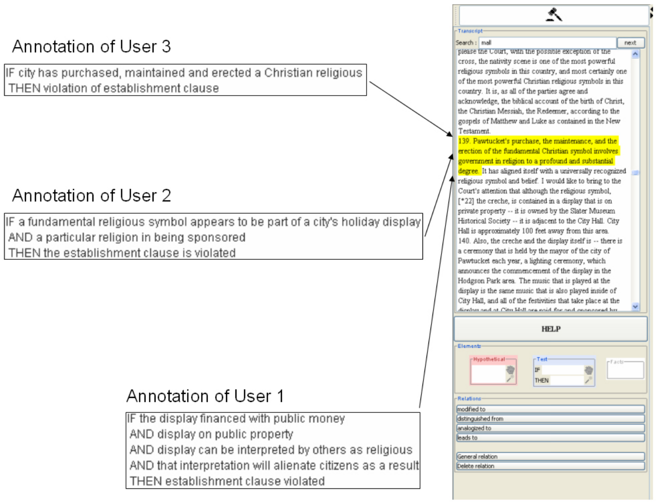

Fig. 1. Transcript annotations: the principle for generating rating dialogs based on markups

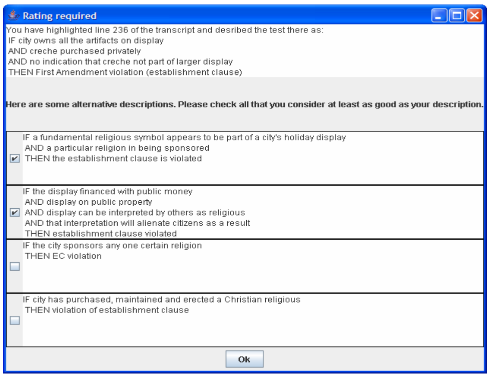

Fig. 2. Example of a rating dialog in LARGO

created a corresponding element in the diagram themselves, users are presented with a small number of alternative answers that were given by other users. They are then asked to select all those they consider of high quality (Figure 1 and 2).

Based on the evaluations a user makes, a heuristic of the quality of his own answer can be calculated based on the assumption that recognizing good answers is an indication of having understood the argument component, which in turn is a prerequisite for having created a good quality contribution oneself. This first heuristic measure is called the *base rating*. The base rating of an answer is immediately available after the user has provided his ratings. It measures to what extent a user can recognize good passage descriptions and thus serves as an initial heuristic of his contribution's quality. But the base rating does not measure the description that the user has actually typed in. Following the collaborative filtering paradigm, LARGO measures this by the positive and negative evaluations that a contribution receives over time by peers.

We call this second measure the *evaluation rating*. Finally, an overall *quality rating* of a contribution can be calculated as the weighted average of the base and evaluation ratings, with the number of received positive and negative evaluations determining the weight of the evaluation rating component.

While this approach works fine for most of the users in the group, the first users who describe on a specific part of the transcript need special attention. For these users, other descriptions that they could rate are not available yet. Here, LARGO uses system provided answers of known quality (some bad, some good) in order to deal with the "cold start problem" of recommender systems. These expert grades ensure a good initial quality heuristic in the system.

The LARGO approach is similar to the reciprocal review system of SWoRD [21], but differs in two respects. First, no textual reviews are required and only quick yes/no decisions are employed within the evaluation questions. The approach was chosen on order not to distract the users from their main activity. The evaluation of peer answers is merely a "social side activity". Another difference to SWoRD and other classical peer review systems is that a rating has immediate implications for the system heuristic about both the *rated* text and also the *rater's* own text. For the rated text, the evaluation contributes to the evaluation rating part of the quality heuristics. For the rater's own text, the evaluation constitutes the base rating. Compared to other recommender systems, the LARGO system is designed also to work with fewer numbers. It is thus more appropriate for cooperative applications with smaller groups.

4 Conclusion

Groupware and Social Software are two terms with different origins that are both used for systems that facilitate social interaction. Given the lack of precise definitions for both terms and the variety of research and development in both fields, a clear borderline between these two overlapping concepts cannot be drawn. This paper argued that there are differences between typical groupware systems and Social Software in terms of application areas, degrees of user control, technology requirements, success factors and algorithms. Yet, these differences are not a strict necessity. To the contrary, they can inform the research and development of CSCW tools by applying design principles that have proven successful in the Social Software field. The LARGO system for legal argumentation training through cooperative visualization of arguments illustrates this by using collaborative filtering, a core Social Software technology, for a serious, goal-driven application.

Acknowledgements

The LARGO system has been developed in joint work with Vincent Aleven (Carnegie Mellon University), Collin Lynch and Kevin Ashley (both University of Pittsburgh).

References

1. Allen, C.: Definitions of groupware. Applied Groupware 1, 1–2 (1990)
2. Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: some issues and experiences. Communications of the ACM 34(1), 39–58 (1991)

3. Marca, D., Bock, G. (eds.): Groupware: Software for Computer-Supported Cooperative Work. IEEE Press, Los Alamitos (1992)
4. Christopher Allen Blog, http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html
5. David, S., Pinch, T.: Six degrees of reputation: The use and abuse of online review and recommendation systems. First Monday 11(3) (2006)
6. BPMN Information, http://www.bpmn.org/
7. IMS Learning Design, http://www.imsglobal.org/learningdesign/
8. Wilson, S., Galliers, J., Fone, J.: Not all sharing is equal: the impact of a large display on small group collaborative work. In: Proc. of CSCW, pp. 25–28. ACM, New York (2006)
9. Fono, D., Counts, S.: Sandboxes: supporting social play through collaborative multimedia composition on mobile phones. In: Proc. of CSCW, pp. 163–166. ACM, New York (2006)
10. Grudin, J., Palen, L.: Why Groupware Succeeds: Discretion or Mandate? In: Proceedings of ECSCW, pp. 85–93. Kluwer, Dordrecht (1995)
11. Neuwirth, C.M.: Collaborative Writing: Practical Problems and Suspective Solutions. ECSCW, Tutorial Notes #11 (1995)
12. Streitz, N., Tandler, P., Müller-Tomfelde, C., Konomi, S.: Roomware: Towards the Next Generation of Human-Computer Interaction based on an Integrated Design of Real and Virtual Worlds. In: Human-Computer Interaction in the New Millenium. Add.Wesley (2001)
13. BSCW Information, http://www.bscw.de/english/index.html
14. Sun, D., Sun, C.: Operation context and context-based operational transformation. In: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, pp. 279–288. ACM, New York (2006)
15. Papadopoulos, C.: Improving Awareness in Mobile CSCW. IEEE Transactions on Mobile Computing 5(0), 1331–1346 (2006)
16. Konstan, J., Riedl, J.: Collaborative Filtering: Supporting social navigation in large, crowded infospaces. In: Designing Information Spaces, pp. 43–81. Springer, Berlin (2002)
17. Verheij, B.: Artificial argument assistants for defeasible argumentation. Artificial Intelligence 150, 291–324 (2003)
18. Aleven, V.: Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment, AI 150, pp. 183–238 (2003)
19. Pinkwart, N., Aleven, V., Ashley, K., Lynch, C.: Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 227–236. Springer, Heidelberg (2006)
20. Ashley, K.: Modeling Legal Argument: Reasoning with Cases and Hypotheticals. MIT Press/Bradford Books (1990)
21. Farzan, R., Brusilovsky, P.: Social Navigation Support through Annotation-Based Group Modeling. In: Proceedings of User Modeling, pp. 387–391. Springer, Berlin (2005)