“Channeling”: Applying Social Software Design Principles to CSCW Scenarios

Niels Pinkwart
Clausthal University of Technology
niels.pinkwart@tu-clausthal.de

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, 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 and CSCW literature shows that the ties are not as strong as one would naively suspect. While there are two Social Software related workshops at ECSCW 2007 and there was one paper session on “Social tagging and recommending” at CSCW 2006 and three on “social computing” at CHI 2007, very little has been before that. It cannot be stated that the origin of Social Software was in CSCW – in fact, the research community rather slowly seems to make connections to the phenomenon of Social Software. But if the Social Software movement was not a result of CSCW research, what are then the relations between the two – or more precisely, between the software tools employed in CSCW (i.e., groupware tools), and Social Software?

Groupware vs. Social Software

At first sight, the aims of both groupware and CSCW seem similar. According to the most frequently used definitions, the term "groupware" denotes software that supports intentional group processes (Allen, 1990) and that serves groups of users with a shared aim or goal (Ellis et al., 1991). This is very similar to the current usage of the term Social Software as a characterization of all kinds of systems that support group interaction. Differences, however, can be found with respect to the following dimensions:

- **Control**: Social Software systems are typically very open in that they delegate a lot of control to the users and the community. For example, Wikipedia entries are generally not "reviewed" or "edited" – the control works largely based on social protocols, supported by some technology (logging entries and keeping entry versions). Also, there is usually little "process control structure" in Social Software systems. This is different in many groupware tools where the system side control about possible user actions is an important factor (e.g., in workflow systems). CSCW tools are often *about* scaffolding the group process – this implies a certain possible intervention in the options of single users.
- **Application Areas:** CSCW and groupware are traditionally oriented towards supporting group *work* and enabling collaborators (i.e., co-workers) to interact productively and efficiently. There is a much greater heterogeneity in Social Software tools in this regard. These also target fields like hobbies, leisure, or play, and consequently the tools are less driven by goals like productivity and efficiency. However, it should be noted that there are indeed work or business related Social Software tools (such as linkedin, a Social Networking Services for professional networking).
- **Used Technologies**: A considerable portion of current CSCW research (and groupware tools) has always been devoted to studies how modern technologies (like today's big displays, PDAs or cell phones) can support group interactions. Compared to that, most Social Software is rather "low tech" and requires not more than a Web access and a browser (cf. next point). The avoidance of expensive or proprietary technology allows many users to access the systems.
- **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. The benefits of the systems are obvious and often also available for non-members. For instance, Wikipedia and flickr are open for everybody, and also the "social aspects" of amazon.com

(recommendations etc.) are visible without even logging in. CSCW tools, on the other had, are frequently tailored towards smaller but better structured groups and group processes. They do not need huge user communities and, 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. This involves aspects of sociology, psychology, economics, computer science, and the specific domain targeted.

- **Algorithms:** CSCW research and the corresponding groupware systems build upon (and develop) a wide variety of algorithms. This includes, for instance, methods for concurrent text editing, algorithms for awareness information, and conflict detection / resolution strategies. For Social Software systems, the one by far most prominent and widely used algorithm is collaborative filtering: through their actions in the system, users get in various ways associated to artifacts. 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 (or users) to other users.

The sequel of this position paper outlines a scenario that shows how some of the differences between groupware and Social Software can be overcome. We illustrate how the most prominent algorithm within Social Software systems – collaborative filtering – can be embedded within a serious, goal-driven CSCW scenario. We call this embedment "channeling" in order to emphasize its closed character: in the scenario, collaborative filtering is employed only as a specific tool at well-calculated spots – not as a general system foundation, as in many Social Software systems.

Applying Social Software principles to CSCW – an Example Scenario

Our example scenario is rooted in the domain of law and in particular the training of legal argumentation skills, which are central for advocates. There are only few systems which support users in the acquisition of these kinds of skills (e.g. Aleven, 2003; Verheij, 2003). One reason for this is that legal argumentation is an ill-defined domain (Lynch et al., 2006) - 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 (Legal Argumentation Graph Observer) system is designed to teach a group of users legal argumentation skills by allowing them to individually analyze examples of expert argumentation (in our case, transcripts of US Supreme

court oral arguments) and then use others' analyses in an embedded recommender system to improve the overall solution quality.

In LARGO, a transcript analysis is done by marking up the text transcript and annotating (tagging) passages with typed descriptions, which can be put in visual relation to each other, thereby forming an argument diagram. The available types for the annotations correspond to an argumentation model (cf. Ashley, 1990 for details). For instance, a "test" represents a decision rule proposed by an attorney, and a "hypothetical" stands for a challenging scenario, posed by a Justice, 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.

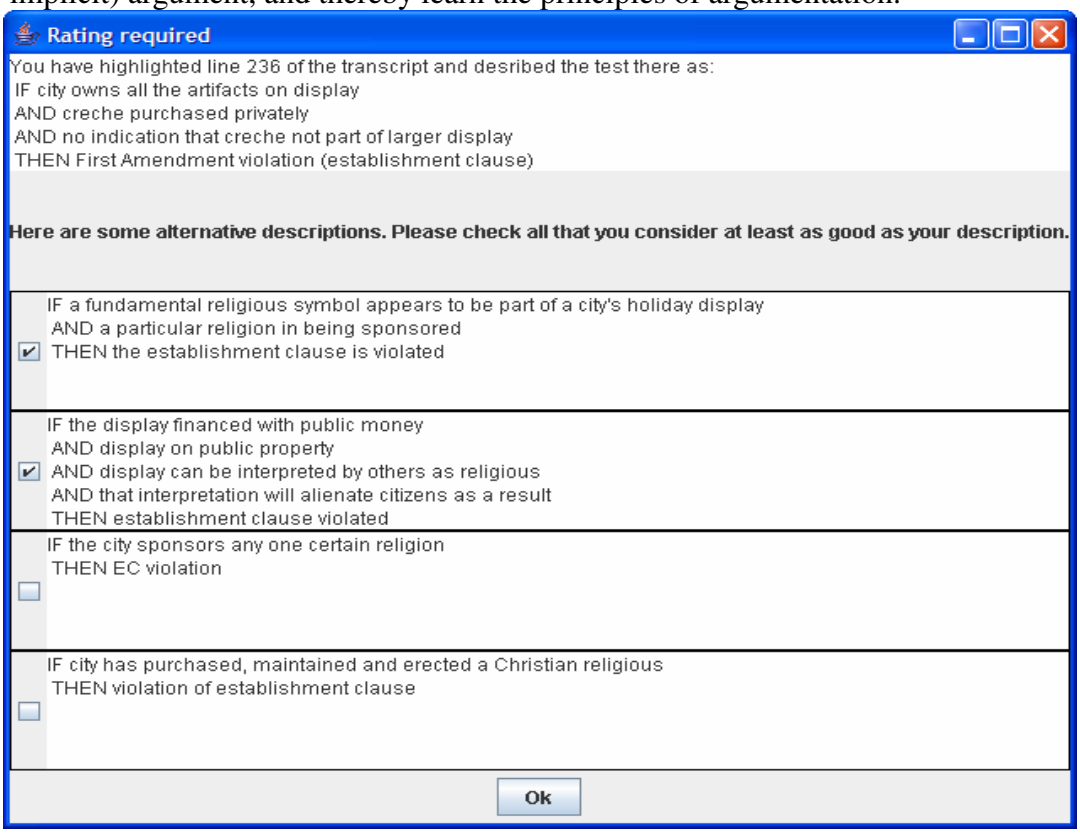

Fig. 1 Example of rating dialog in LARGO

LARGO analyzes the structure of user-created argument diagrams and gives feedback on it (cf. Pinkwart et al., 2006, for details about the system feedback). System feedback is intended to help users create good argument structures that are related to the transcript markups in a reasonable way. Yet, users may have difficulties in understanding, e.g., the essence of a proposed test, as evidenced by a poor paraphrase in the corresponding test node they add to the diagram. Obviously, this is very hard to detect by the system, since it involves

interpretation of legal argument in 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 matches the content. For a computer program it is certainly not easier to do this. The tags and markups together with other group members working on the same task can help here, since this combination enables a quality heuristic for single argument components (such as a test description) based on collaborative filtering (Konstan & Riedl, 2002). In our 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 created a corresponding element in the diagram, users are presented with a small number of alternative answers (given by other users) and asked to select all those they consider of high quality (cf. figure 1 and 2).

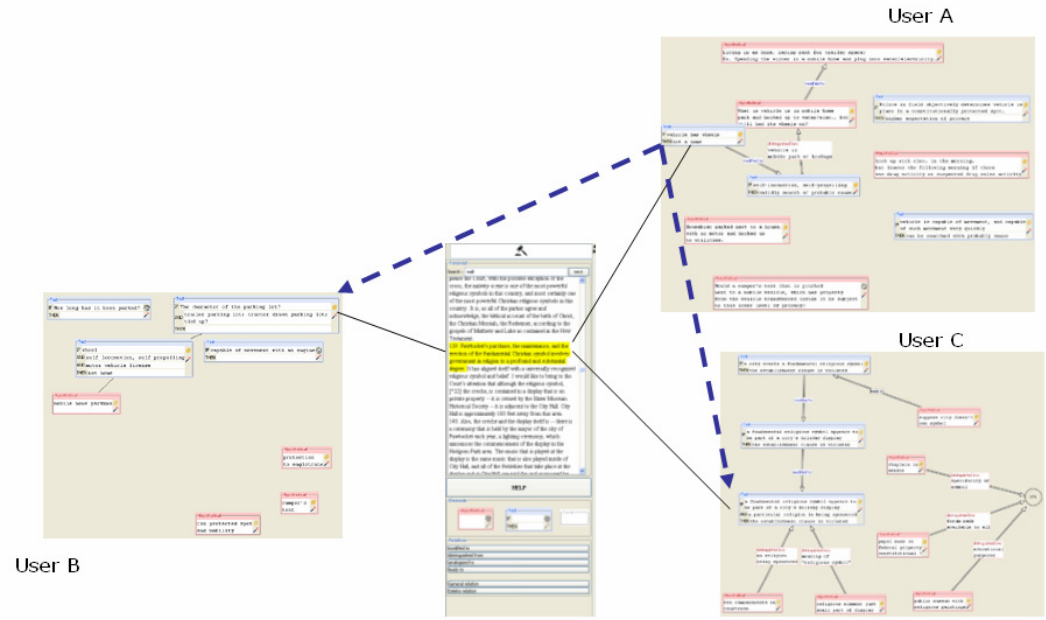

Fig. 2 Principle for generating rating dialogs based on markups and individual descriptions (tags) in diagrams

Based on the evaluations a user makes, a first heuristic of the quality of his own answer can be calculated. One may assume 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. We call this first heuristic measure the *base rating*. The base rating of an answer is immediately available after the user has provided his ratings. It measures in how far a user can recognize good passage descriptions and thus serves as a heuristic of his contribution's quality, but does not rate the description the user has actually

typed in. Following the collaborative filtering idea, this can be measured by the positive and negative evaluations that a contribution receives. We call this 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 work on a specific part of the transcript (and thus are the first to markup and tag it and subsequently evaluate other descriptions) need special attention. For the first users that annotate a specific passage of the text, other descriptions are not available yet. Here, we use system provided answers of known quality (some bad, some good) in order to deal with the cold start problem. These expert grades ensure a good initial quality heuristic in the system.

The LARGO approach is similar to the reciprocal review system of SWoRD (Farzan & Brusilovsky, 2005), but differs in three respects. First, no textual reviews are required and only quick yes/no decisions are employed within the evaluation questions. The approach is geared towards not distracting the user from his main activity and includes the evaluation of peer answers as a "social side activity". Another difference to SWoRD and other classical peer review systems is that 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 feeds into the evaluation rating part of the quality heuristics, and for the rater's text, the evaluation constitutes the base rating. Finally, a difference to SWoRD is that the object of rating is of finer granularity – while SWoRD uses larger writing samples, our approach is based on very small annotations of a specific part of a resource (i.e., the argument transcript). This probably helps integrating user's own analysis activity with the evaluation activity, since the thematic proximity of own work and the statements to be evaluated is likely to be very close. Compared to other recommender systems, which essentially rely on large user group sizes, our system is designed also to work with fewer numbers (through the inclusion of the base ratings) and thus more appropriate for small group work scenarios.

Conclusion

After a discussion of typical differences and similarities between CSCW/groupware and Social Software, this paper presented an example scenario that shows how key principles of Social Software can be used within CSCW research and groupware design. The example scenario from the domain of legal argumentation illustrates how the collaborative filtering algorithm and user activities like markup and tagging, all characteristics for Social Software, can be applied and "channeled" into a goal-oriented, serious application which uses an indirect collaboration mechanism (peer rating).

References

Aleven, V. (2003): 'Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment', *Artificial Intelligence,* vol. 150, pp. 183-238.

Allen, C. (1990): 'Definitions of groupware', *Applied Groupware*, vol. 1, 1990, pp. 1-2.

Ashley, K. (1990): *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge MA, MIT Press/Bradford Books.

Ellis, C. A., Gibbs, S. J. and Rein, G. (1991): 'Groupware: some issues and experiences', *Communications of the ACM*, vol. 34, no. 1, pp. 39-58.

Farzan, R. and Brusilovsky, P. (2005): 'Social Navigation Support through Annotation-Based Group Modeling' In Proc. of User Modeling, pp. 387-391. Berlin, Springer.

Lynch, C., Ashley, K., Aleven, V. and Pinkwart, N. (2006): 'Defining Ill-Defined Domains; A literature survey' In V. Aleven, K. Ashley, C. Lynch and N. Pinkwart (eds.), *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems* (p. 1-10). Jhongli (Taiwan), National Central University.

Konstan, J. and Riedl, J. (2002): 'Collaborative Filtering: Supporting social navigation in large, crowded infospaces' In *Designing Information Spaces: The Social Navigation Approach*, pp. 43-81. Berlin: Springer.

Pinkwart, N., Aleven, V., Ashley, K. and Lynch, C. (2006): 'Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques' In M. Ikeda, K. Ashley and T. W. Chan (eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science 4053* (p. 227-236). Berlin (Germany), Springer.

Verheij, B. (2003): 'Artificial argument assistants for defeasible argumentation', *Artificial Intelligence,* vol. 150, pp. 291-324.