Social Adaptation of ERP Software: Tagging UI Elements

Marcus Nitsche[1], Martin Christof Kindsmüller[2], Udo Arend[3], and Michael Herczeg[2]

[1] Institute for Technical and Operational Information Systems, Otto-von-Guericke-University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany
marcus.nitsche@ovgu.de

[2] Institute for Multimedia and Interactive Systems, University of Lübeck
Ratzeburger Allee 160, 23538 Lübeck, Germany
{mck|herczeg}@imis.uni-luebeck.de

[3] SAP AG
Raiffeisenring 45, 68789 St. Leon-Rot, Germany
udo.arend@sap.com

Abstract. In this paper we present a newly designed annotation and collaboration component, which has been prototypically implemented on top of an ERP (Enterprise Resource Planning) system. Standard ERP software is often described as being inflexible in respect to personal needs of a single user or a user group. To cope with this problem, theories, and principles from classical *CSCW* research and design were combined with recent developments in *Social Software* and what is now often summarized as *Web 2.0*. Our component is inspired by Web 2.0 principles like user generated content, information sharing, and harnessing network effects. As the central paradigm we applied *social tagging* based on *folksonomies* as e.g. used in del.ico.us, Flickr, and YouTube. In addition best practices from research on *online community building* were used to design a *social annotating component* for ERP systems.

Keywords: Human-Computer Interaction, Social Software, Tagging, Social Bookmarking, Social Annotating, Enterprise Resource Planning

1 Introduction

Enterprise resource planning (ERP) software consists of powerful applications for managing company resources like human resources, finances, and other means of production in order to support effective and efficient business processes. But often they are described as too complex and too inflexible in case of user's needs (cf. [1]). Contrary to this situation Web 2.0 applications are often regarded as being the opposite: People like to use Flickr, Facebook, YouTube, and other information sharing platforms. So the idea came up to enhance the usability and the joy-of-use of an ERP system by applying principles known from social software [2].

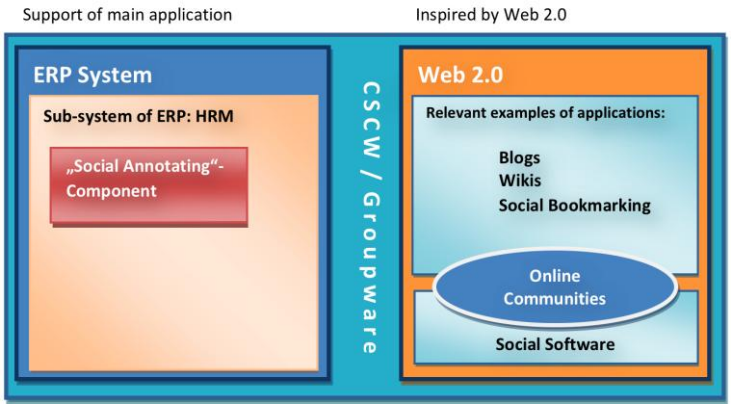

Fig. 1. Broad field of development and theoretical basics [8]

As shown in Fig. 1 on the right side aspects of social software, online communities [3], and relevant examples of Web 2.0 applications were studied. As promising for enhancing an ERP system, we decided for a social annotating component. To illustrate the power of our approach we decided for using Human Resource Management (HRM) domain with the task "Adding a new employee".

Following O'Reilly's [4] definition of Web 2.0 as "…the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform" and considering O'Reilly's statement that the most important of these rules is "… build applications that harness network effects to get better the more people use them" [4] it is obvious that Web 2.0 often needs a community to support these network effects [3].

Second, the so-called *social software* can be defined like in the following [5]: "Social software enables people to connect and collaborate through computer-mediated communication. It also enables people to form online communities". According to this definition the building of online communities is a possible step but not necessary to benefit from the proposed social annotating component [3]. In the following chapter we describe how to apply those principles to such a component.

2 A Social Annotating Component

The concept of social annotating is based on two aspects. First, people love to communicate with each others. That is one of the reasons why Web 2.0 applications are so popular. Not even bad interface design or limitations in the communication channels can stop people from communicating via computer systems (as indicated by early research on computer mediated communication, e.g. [6]). Second, people like to keep their environment consistent. Therefore users like to adapt software to their personal needs whenever they are able to do so.

The idea is to adapt social annotating techniques to account also for manipulating certain components on the user interface (UI). Through this technique users of an ERP system are able to share their experience and insights with other users. Based on our assumptions we state the hypothesis that user-annotated manipulations of elements of the UI lead to improvements of the user interaction in performing business processes.

Some remarks on annotations: "We annotate data all the time. When we read a paragraph, and mark it 'great!' in the margin, that is an annotation. When our text editor underlines a misspelled word, that is also an annotation" [6]. Therefore annotating is adding meta-information to the existing information by marking, commenting, or sketching something.

Giving the user the opportunity to annotate work objects (e.g. entry fields in form based interfaces), to change its properties, and to keep their adaptations in a close context, may influence the way how people perform their tasks. In principle also adaptations of other user interface components are possible. In any case of annotating a screen object, the user is asked to add a comment to communicate to other users why a specific adaptation has been made. Whether these annotations and adaptations are of permanent character is decided by the whole community via a rating process.

Community feedback can be used according to the principles of information sharing to create a context-close communication opportunity between users. Information on personal notes which were placed previously on users' desks or monitors could be useful for other users as well.

Social software is often only useful for a single user if enough other people are already using it. The proposed tool does not suffer from this chicken-and-egg problem as it incorporates a "*killer feature*" [3]: A single user can benefit from using this tool, even with nobody else using it. In our case the tool can act as context related external memory with personalization function. Two user modes are supported implicitly (Fig. 2). The user can switch between these modes by simply setting the view restrictions to *public* or *group* rather than *private*.

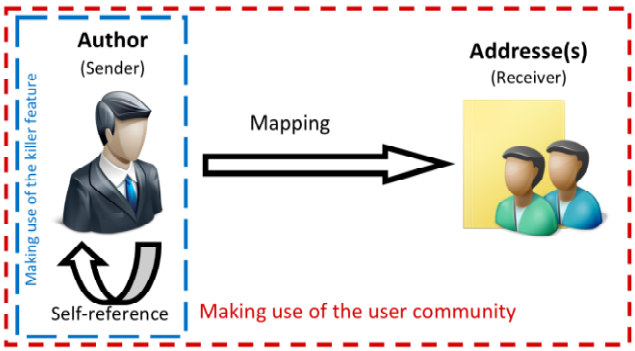

Fig. 2. Basic use modes of the social annotating component [8]

Before we present the concept and implementation of the prototype we want to take a closer look at the cause-and-effect chain. Three crucial questions in regard to *cause*, *relation*, and *effect* need to be answered.

1. *Cause*: Annotations are caused (created, commented, and rated) by the end user.
2. *Relation*: *A*nnotations are related to static business objects as well as to dynamic business processes. For each annotation the user can choose whether it is *process-* or *object-related* (in our example "adding a new employee" is the *business process* and the employee the user likes to add to the system is the *business object*).
3. *Effect:* Each annotation affects the system. Depending on the type of annotation *synchronous* or *asynchronous effects* are possible. Synchronous reactions are triggered by critical issues like "important document is missing, please hand in soon". Asynchronous effects like reading an old comment from months ago can remind the user to be aware of something he or she should not forget when dealing with this business process.

According to a user survey [8] among employees in human resources (HR) the following annotation types are supported by the system: (1) *Comments*, (2) setting *Default Values*, (3) *Hide*/*Unhide Fields*, (4) *Enhancements*, (5) *Best Practices*, and (6) *Exceptions*.

Each type supports the direct annotation of each form object and therefore creates context proximity. The annotation data will be saved together with the business object "employee". Alternatively the annotation data can be saved for the whole business process "Add employee" (cf. "2. Relation"). In the second case depending on chosen view restrictions the user or the whole community will see the annotation and its effects whenever an employee will be added in the future. While a business object-related annotation affects only the special employee the user like to add, the business process-related annotation affects the whole process. In the next chapter we present the chosen interaction design in more detail.

3 Interaction Design

Based on different end user studies with HRM employees and several reviews by user interface designers and solution managers [8], we decided for directly annotating UI elements like entry fields by the user to allow for different types of annotations including the personalization of the user interface itself. Therefore we first created paper mockups like the one shown in Fig. 3 and sketched first interaction sequences to discuss possible forms of interaction (Fig. 4). To ensure that the users understand that and why the UI was modified, UI adaptations by the end user can be accompanied by comments.

We followed the hypothesis that adding an annotating component with additional personalization capabilities to an ERP system would enhance the usability, as well as the user experience of the system. Our prototype allows the user to stick textual annotations to program screens of a single instance of a business object, to an activity within a business process, or even to an entire business process. Thus by equipping a user with a reminder or personalization tool we expect to generate immediate benefit for that single user (killer feature according to [3]). The component was designed to allow users to publish information about business processes in a direct contextual way using drag-and-drop of virtual notes in a direct manipulation paradigm.

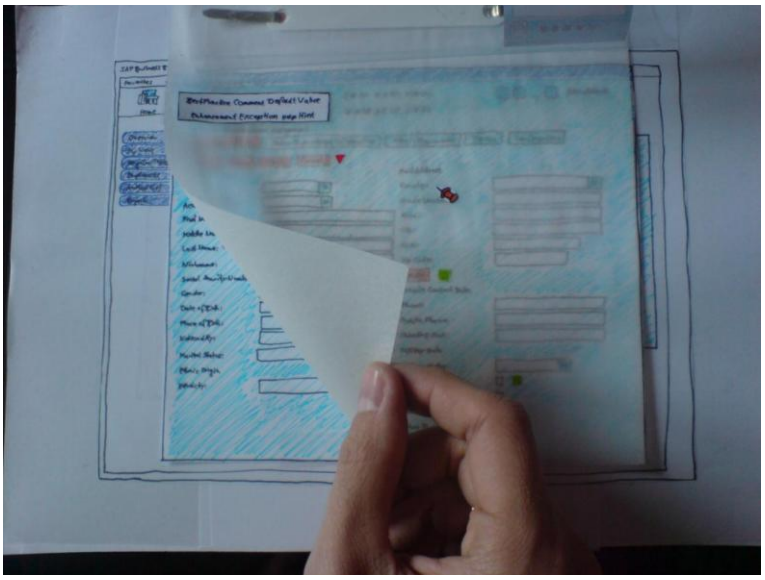

Fig. 3. Paper mockup of a first prototype [8]

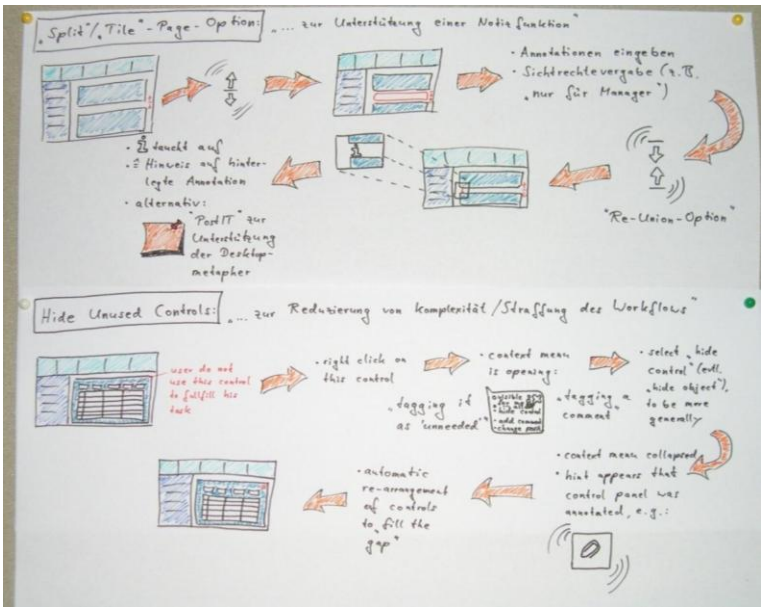

Fig. 4. Interaction sequences of early concepts [8]

Furthermore the user is able to do privacy adjustments in order to restrict the visibility of the annotation and to classify an annotation as “business process related”.

After sketching the application a prototype was created and evaluated. This is explained in the next two sections.

4 The Prototype and an Exemplary Workflow

First, the prototype was supposed to be implemented as a high-level mockup (e.g. as HTML pages or PowerPoint slides). But because of the amount of interactions we decided for Adobe's Flex 3 framework, which can be used to create rich internet applications, as well as basic prototypes of web applications as used here.

Technically, the frontend consists of a flex client, which is the users' interface to exchange application data with the backend (*Apache* web server). All meta-information (annotation data sets of different types consisting of e.g. ID, position/context, content, category, view restrictions, etc.) are stored in a MySQL database.

To stay consistent with the existing HRM application we rebuilt the relevant part of the application in Flex and added our social annotating component in the right upper corner of the "Add Employee"-window (Fig. 5). After dragging the yellow note to the user's point of interest, a context sensitive annotating console pops up and allows the user to fill in annotations of each supported kind, in order to adapt it to personal needs like already explained in previous chapters.

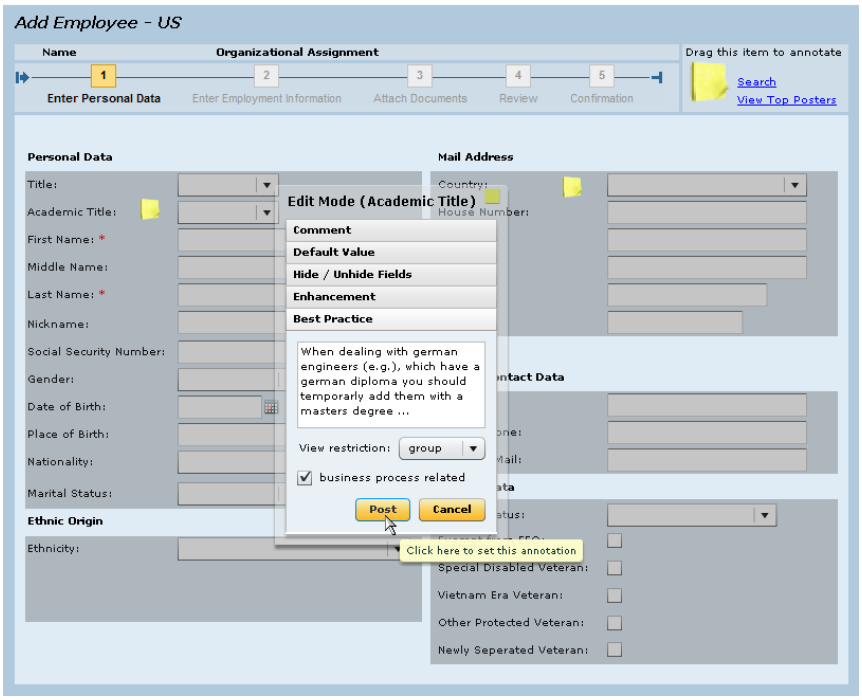

Fig. 5. Screenshot of a phrased annotation, ready to post. [8]

As an exemplary workflow, we present the interaction sequence the user needs to perform to set a personalized *Default Value* to "Academic Title" (cf. Fig. 6):

1. The user drags the yellow note icon from the right upper screen corner to "Academic Title" (cf. Fig. 6, step 1). If he or she drops the note for example in the proximity of the label or the corresponding entry field of "Academic Title" the annotating dialog appears.

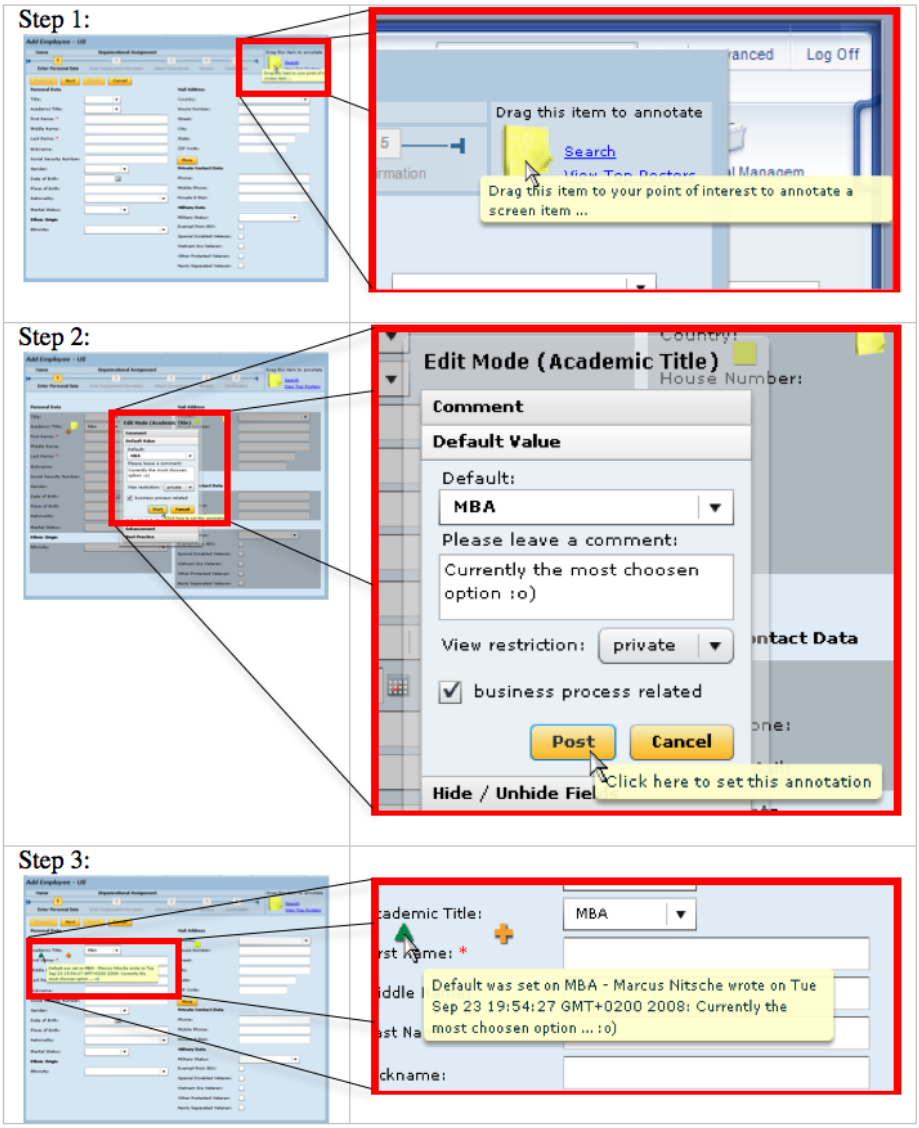

Fig. 6. Form screen and detail view during adding an annotation to "Academic Title". The rectangle shows the detail view in the form [8].

2. Now the user needs to choose which kind of annotation he or she likes to set. In our case it is "Default Value" (cf. Fig. 6, step 2).
3. In the upcoming page of this accordion menu [9] the user is asked, (1) to set the default value he or she likes to see automatically in future, (2) to enter a comment, (4) to set the view restrictions, and (4) to decide whether this annotation is business process related or only applies to the case currently under examination (cf. Fig. 6, step 2).
4. Finally the user needs to click on "Post" to set the annotation.

After these steps an icon (in case of a "set default value" it is a green triangle) appears in proximity of the label "Academic Title" indicating that this type of annotation was set here. Hovering over this icon presents the corresponding author, date, and comment to this annotation (cf. Fig. 6, step 3).

The tool has been evaluated with several usability experts as described in the next section.

5 Evaluation

The development process was accompanied by formative evaluation steps to analyze and improve the usability as well as the user experience of the prototype [10, 11]. According to the stages, presented in (Fig. 7), different methods (e.g. expert user testing, think-aloud protocols, post-use satisfaction survey [12]) were used in an iterative evaluation process [13].

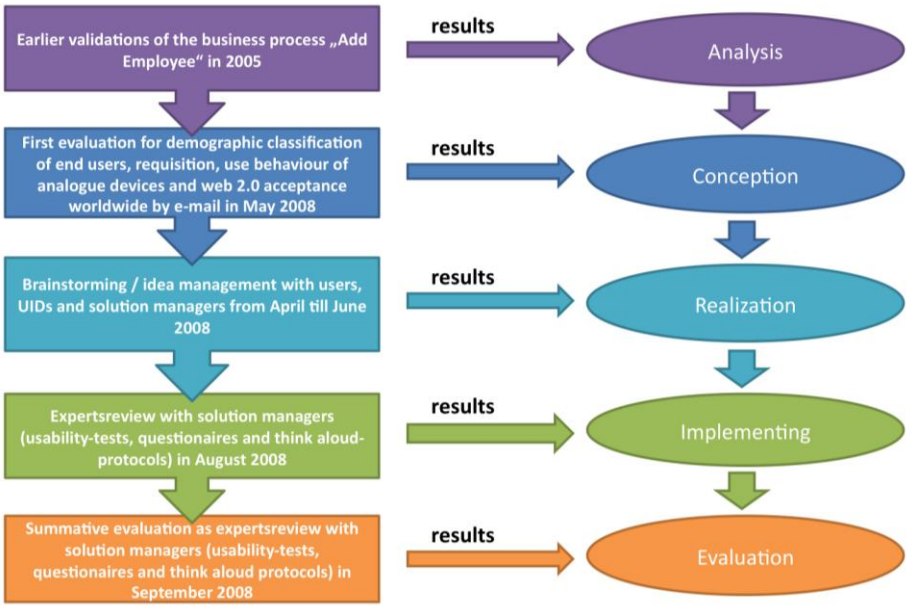

Fig. 7. Formative evaluation process by multiple user feedback [8, slightly adapted]

Finally, a summative evaluation with five solution managers as participants has been conducted. These business experts were asked to answer usability related questions on a 7-step Likert rating scale from 1 (very bad) to 7 (very good). The results of the final post-use satisfaction survey can be read up in Table 1.

Table 1. Results of post-use satisfaction survey of summative end evaluation with five solution managers (SD = standard deviation) [08]

Question	SD	Mean
1. Overall, how useful would this application prototype be for you to perform your job?	0.632	**5.0**
2. Overall, how much functionality was missing from the prototype that you would need to do your job?	0.632	**6.0**
3. Overall, how easy or difficult was it for you to perform tasks today?	1.020	**5.6**
4. Overall, how attractive was the user interface?	0.490	**6.6**
5. Overall, how consistent was the user interface?	1.356	**5.6**
6. Overall, how well did the terminology presented in the prototype match your needs?	1.414	**5.0**
7. Overall, how satisfied are you with this application prototype?	0.748	**5.8**

The results of this test of the social annotation component revealed a high degree of usability and a high acceptance of this feature. Attractiveness and functionality has been rated high. The adequacy of the terminology has been rated lower. As the participants stated, that they would use these features in practice, they might become even more convinced of the usefulness.

6 Conclusions

According to our evaluation results combining tagging and personalization is a promising approach to enhance usability of a business application in applying Web 2.0 techniques. It still has to be evaluated if business users accept the functionality in real life situations. As the proposed tool employs a "*killer feature*" in the sense of [3], a user can start using the system simply as a kind of personal replacement for post-its, which tend to clutter up his or her computer screen or desk. At this time the *Social Annotating Tool* is not social at all, but nothing more than an intelligent collection of post-its that are context aware.

Over time some users might become willing to share personal insights and experiences with other colleagues. Because all users become instantly aware as soon as some start sharing, this can easily cause a snowball effect. This way community building [3] can be enhanced, because the first stages and levels of participation in the online community membership lifecycle: *visitors*, *novices*, and *regulars* [14] are very well supported by the tool. If this can be empirically confirmed, it would mean that colleagues could benefit from the *Social Annotating Tool* in optimizing their own interaction behavior.

7 Acknowledgement

Marcus Nitsche is funded by the German Ministry of Education and Science (BMBF) within the ViERforES project (no. 01IM08003C).

References

1. Koll, S.: Firmenweite Standardsoftware steht am Komplexitätspranger. Computer Zeitung 38, p. 10. (2007)
2. Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S., Yoshida, S.: Socialware: Multi-agent systems for supporting network communities. Communication of the ACM, 42(3), 55-61 (1999)
3. Kindsmüller, M.C., Melzer, A., Mentler, T.: Online Communities and Online Community Building. In: Khosrow-Pour, M. (ed.) Encyclopedia of Information Science and Technology, pp. 2899--2905. Information Science Publishing, Hershey (2009)
4. O'Reilly, T.: Web 2.0 Compact Definition: Trying Again. 10.12.2006, downloaded 05/15/2008 from O'Reilly Radar: http://radar.oreilly.com/archives/2006/12/web-20-compact-definition-tryi.html (2006)
5. Koskinen, T.: Social software for industrial interaction. ACM International Conference Proceeding Series; Vol. 206, Proceedings of the 20th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artefacts and environments, pp. 381--384. ACM, New York, NY (2006)
6. Turoff, M., Hiltz, S. R., Kerr, E. B.: Controversies in the design of computer-mediated communication systems: A Delphi study. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Gaithersburg, Maryland, United States), pp. 89-100. ACM, New York, NY (1982)
7. Oren, E., Delbru, R., Völkel, M., Handschuh, S.: Annotation and Navigation in Semantic Wikis. ESWC2006: Workshop on Semantic Wikis, downloaded 07/02/2008 from semwiki.org: http://semwiki.org/semwiki2006 (2006)
8. Nitsche, M.: Nutzerzentrierte Gestaltung einer "Social Annotating"-Komponente zur Unterstützung von Business Prozessen in ERP-Software. Diploma Thesis, Institute for Multimedia and Interactive Systems, University of Lübeck in cooperation with SAP AG, Walldorf / St. Leon-Rot (2008)
9. Buyukkokten, O., Garcia-Molina, H., Paepcke, A.: Accordion summarization for end-game browsing on PDAs and cellular phones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Seattle, Washington, United States), CHI '01, pp. 213-220. ACM, New York, NY (2001)
10. Cato, J.: User-Centred Web Design. Harlow, UK: Addison-Wesley, pp. 191--213 (2001)
11. Shneiderman, B., Plaisant, C.: Designing the User Interface. Harlow, UK: Pearson Addison Wesley (2005)
12. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction, (3rd edition), Prentice Hall, Harlow, UK (2004)
13. Nielsen, J., Landauer, T. K.: A mathematical model of the finding of usability problems. In: Proceedings of ACM INTERCHI'93 Conference (Amsterdam, The Netherlands, 24-29 April 1993), pp. 206--213.
14. Kim, A. J.: Community Building on the Web: Secret Strategies for Successful Online Communities. Peachpit Press, Berkeley, CA (2000)