Lightweight Support for Collaborative Web Browsing Through SpreadVector

Mirko Fetter, Ralf Strobel, Tom Gross
Human-Computer Interaction Group
University of Bamberg
96045 Bamberg
<firstname.lastname>(at)uni-bamberg.de

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2581324.

Abstract
We present SpreadVector, a prototype providing lightweight support for collaborative Web browsing. While a broad need for co-browsing tools has been repeatedly identified, the rate of mainstream adoption for existing solutions is still marginal. We argue that one reason is the lack of tools that offer lightweight collaborative browsing support. With SpreadVector we provide a concept and prototype for lightweight co-browsing.

Keywords
Collaborative Web Browsing; Social Search; XMPP.

ACM Classification Keywords
H.5.3 [Information Interfaces and Presentation]: Group and Organisation Interfaces — Computer-Supported Cooperative Work.

General Terms
Human Factors.

Introduction
While the Web is becoming more and more social, navigating through the Web it is still largely treated as a solitary act by developers of Web browsers. Despite the fact that this has been identified and discussed in detail by researchers before, the number of tools supporting collaborative Web browsing—in either remote or co-located settings—is still very limited. Evans and Chi [1]

identified in their *model of understanding social search*, a variety of forms of social interaction before, during, and after Web searches and derived design suggestions for better tool support. Morris [5, 6] repeatedly studied collaborative Web search practices and found an increasing prevalence of *collaborative search* behaviour, but also confirmed the limited technological innovation to support collaborative information seeking. Based on her findings, she claimed that new solutions needed to be *lightweight.* This point is summarised by one of the respondents in her survey that she quotes as follows: "it might have helped if … if posting something … was faster and required fewer mouse clicks than copying a link into an e-mail message" [6, p. 1189].

In the following we present the concept and interaction design of SpreadVector for the *lightweight* support of collaborative Web browsing. We report on related work, give an overview of the implementation, discuss the results of informal user study, and draw conclusion.

Concept

In the following we describe our concept of *lightweight collaborative Web browsing* support. Our notion of *collaborative Web browsing* primarily aims at short, simple, and unstructured tasks that evolve around visiting Web pages together and communicating in their context—a conception that stresses the requirement of *lightweight* interaction. While various similar notions exist—from *social search* [1] to *collaborative information seeking* [6]—those definitions often resonate with a notion of structure and directedness of the underlying tasks, that we deliberately wanted to bracket off for this work.

In order to distil our concept for the *support of lightweight collaborative Web browsing* we analysed studies and previous work that on opportunities to improve the co-browsing experience such as by lowering the barrier to entry, providing awareness, offering means for communication, or reducing the duplication of effort [1, 3, 5, 6]. We derived three core requirements for SpreadVector to provide *lightweight support*: *lightweight interference* (i.e., minimal change to existing interaction practices); *lightweight transitions* (i.e., seamless switching between browsing alone and browsing together); *lightweight adoption* (i.e., minimal effort for first time usage).

Lightweight interference: In order to minimise the interference with the existing Web browsing practices of the user, we weaved SpreadVector into a standard browser instead of building a separate tool, thus allowing each user to work with their familiar settings, configurations and learned workflows. In order to achieve this while at the same time allowing for tight collaboration between users, one technical challenge had to be faced—that is, rendered presentations of the same Web page, in the same browser, on two computers can greatly diverge for multifarious reasons (e.g., different layouts due to varying resolutions, display and window sizes, font settings, or underlying operating systems as well as due to personalised or dynamic Web content). In collaborative settings, these circumstances make it difficult for users to check if remote users are seeing the same content or to direct the focus of remote users to a specific position in a document. While related work made use of more heavyweight synchronisation approaches like proxies [4] and video streams [7] to support collaboration mechanisms like telepointers, SpreadVector uses an elegant combination of heuristics and the exchange of hash signatures between clients to allow for robust annotation positioning. This way SpreadVector allows users to remotely highlight selections and share annotations between Web browsers on the same Web page.

Lightweight transitions: A further aim of the SpreadVector concept was to allow easy and seamless transitions between solitary and co-browsing activities. Foremost, this includes the possibility to easily initiate a co-browsing session at any time from any given URL during solitary browsing without any "duplication of effort" [6]. Therefore, a user can simply double-click in the SpreadVector GUI on an online contact to initiate a co-browsing session on the current page including an online chat and shared annotations. Vice versa, each participant of a co-browsing session is able to freely switch from the tight collaboration on the same Web page to a more loose collaboration, by simply navigating to a different Web page. In this case, several awareness features—detailed in the next section—allow the user to stay informed over the others' activities, and easily allow transitioning back to a tighter form of collaboration.

Lightweight adoption: Finally, SpreadVector aims to minimise the effort for *first time use* with respect to the installation and first setup in order to lower the entry barrier. SpreadVector makes use of the users' existing Instant Messaging (IM) account to eliminate the need for an upfront registration. This approach also provides the users with a list of potential collaboration partners in form of their existing IM contacts. After two clicks for installing SpreadVector as a Firefox extension SpreadVector can immediately be used. When SpreadVector is opened for the first time, a dialogue asks the user for an account at an existing XMPP service (e.g., Facebook Chat or Google Talk) [9]. Following this step the buddy list is automatically loaded and the first collaborative Web browsing session can be started by clicking on any online contact that is also using SpreadVector. Finally, SpreadVector combines well-known interaction concepts (i.e., a standard browser interface extended with IM-like functionalities) with a carefully designed set of novel co-browsing mechanisms (i.e. shared highlighting and awareness features) to provide high learnability.

Interaction Design

In the following the interaction design of SpreadVector is outlined along the screenshot in Figure 1, with the numbers referring to the items depicted there.

Anytime a need for co-browsing arises—while solitary browsing a Web page (1) in the *content view*—users can easily start SpreadVector by clicking on the SpreadVector icon in the status line (2). SpreadVector automatically logs in with the stored credentials and the appearing *contacts sidebar* (3) provides an overview of the users' contacts in form of a buddy list.

A status message (4) provides awareness over all online contacts and their currently visited Web page in form of a clickable link. By double-clicking on a contact's name, a co-browsing session is initiated and a *co-browsing panel* is created (5) at the bottom of the *content view*, offering a possibility for communication via instant text chat (6). By selecting text on a Web page while holding the ALT-key (7), users can highlighting text to other users. The text is automatically posted as a clickable link in the *co-browsing panel* (8) and the annotation becomes visible in the *content view* of the co-browsing partner.

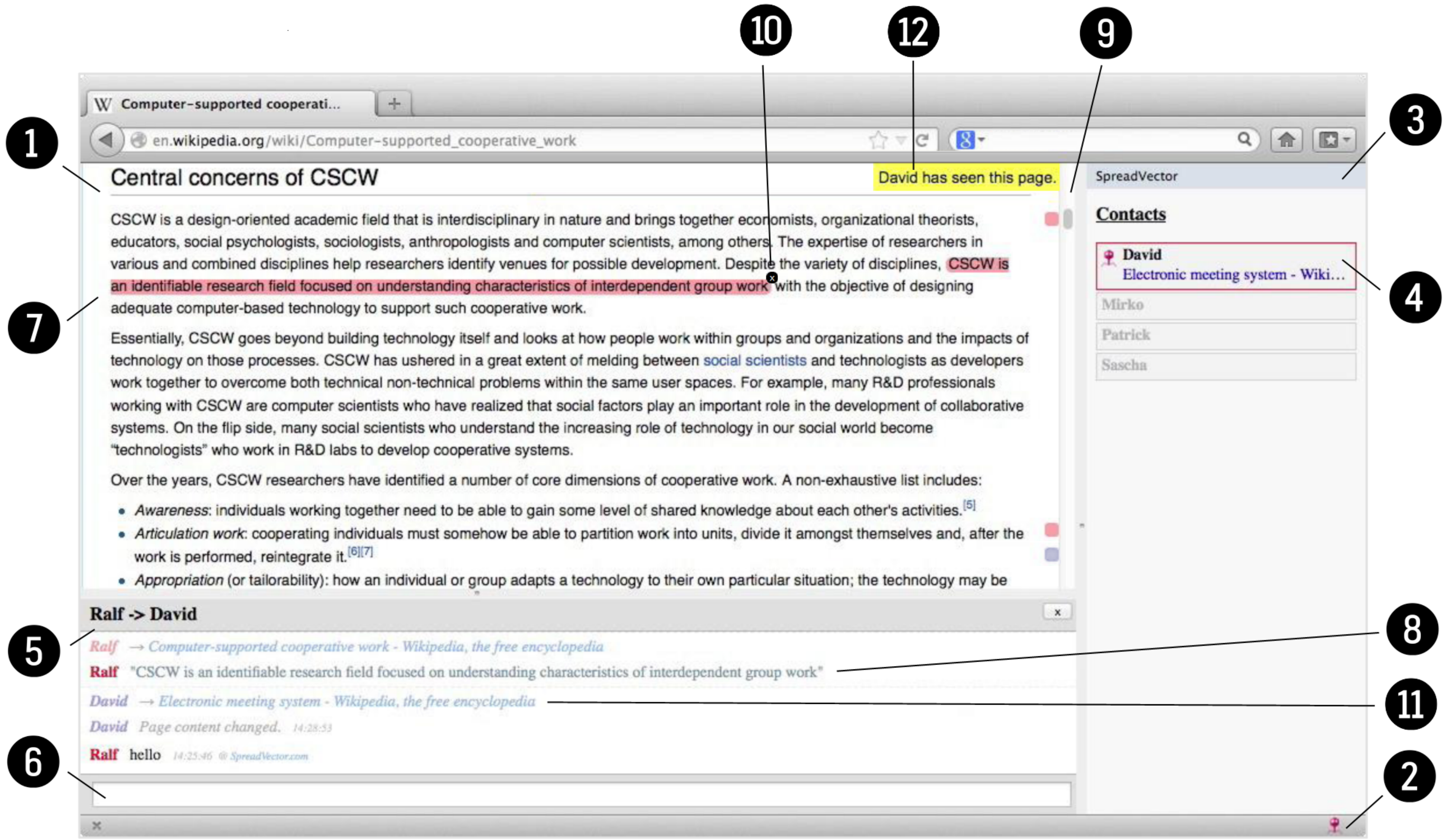

Figure 1. Screenshot of a co-browsing session with the SpreadVector extension in the Firefox browser.

An annotated scrollbar (9) helps to provide an overview of shared annotations on longer documents through scrollbar markers. Different colours allow distinguishing the respective user's own from others' annotations. By hovering over a shared annotation, an appearing 'x' (10) allows to remove annotations. Browsing to a different Web page results in status update in the *contacts sidebar* and is automatically posted into the *co-browsing panel* as a clickable link to provide awareness (11). A further awareness indicator in the top left of the *content view* (12) informs the user with respect to the currently visited page if the co-browsing partner "[Username] has not seen this page", "[Username] is looking at this page", "[Username] is looking at a different version of this page", "[Username] has seen this page", and "[Username] has seen a different version of this page".

Related Work

Research on co-browsing support has started with the advent of the Web. For example, *ComMentor* [8] offered basic support for shared annotations on Web documents. *GroupWeb* [2] is a standalone co-browsing tool, built on top of *GroupKit,* allowing shared textual annotations on Webpages that are visited together and presented in relaxed WYSIWS views. While both solutions provided innovative concepts, the modern WWW—with its graphic intensive layouts, and personalised and dynamic Web contents—poses new technical challenges that those solutions not had to cope with.

Different technical solutions have been developed to address such challenges. For example, *CoFox* [7] sends videos streams of screen captures to remote users to allow for co-browsing. Others developed sophisticated mechanisms [4] using proxies to synchronise dynamic Web content for several users in order to allow co-browsing of dynamic Web content. However, all these technically more heavyweight solutions come with drawbacks for the users.

Also several commercial co-browsing solutions are available, but none of them addresses *lightweight support*. Most of these solutions (e.g., *unblu*, *LiveLOOK* and *Firefly*[1]) aim at enabling agents of commercial Web sites to connect to a customer's browser view (e.g., for remote support or sales) and require an installation on the Web server. The *COVU* browser extension and the standalone application *samesurf*[2] can be seen as the solutions closest to SpreadVector. However, both only allow for tightly coupled master-slave browsing, and do not offer the freedom of SpreadVector. Finally, also off-the shelf screen sharing and remote access tools (e.g., *Skype*, *VNC*) can be repurposed for co-browsing. While they to not offer the comfort of a dedicated solution, they are often used, as they are ready available.

[1] http://www.unblu.com, http://www.livelook.com/, and http://usefirefly.com/

[2] http://www.covu.com/ and http://www.samesurf.com/

Implementation

SpreadVector is implemented as an extension for the Firefox browser to allow for a widespread distribution. The extension is completely written in JavaScript to allow the easy integration into the browser's GUI and Web content. For all communication the widespread protocol XMPP [9] is used, hence allowing users to leverage on existing XMPP accounts (e.g., Google Talk or Facebook Chat) and not requiring additional servers or services. Our implementation adheres to all XMPP relevant standards and thus also allows interaction with regular chat applications. Awareness functionalities and shared annotations are based on observing the DOM tree for mutations and exchanging and comparing hash signatures (e.g., of the annotated text as well as of surrounding elements and text).

User Study and Results

In order to get first insights in the usability of SpreadVector and to test the robustness of the underlying mechanisms with real users, we conducted an informal study during our lab's open house. 17 participants (4 female, 13 male) explored SpreadVector in dyads or triads on one Mac OS X and two Windows computers. After a short introduction, users freely tested the capabilities of SpreadVector for several minutes, and answered a short questionnaire afterwards.

Observation of the users revealed that most of them quickly understood and used the main features of SpreadVector. The informal feedback during this hands-on-phase was very positive as several participants stated that they would like such a tool. However, an issue often addressed during use was privacy. The fact that every page visited is published to all roaster contacts and that all page accesses are listed in the chat history of an active session were disliked by some users.

In the questionnaire 15 participants positively mentioned the highlighting function and the ability to jump directly to a shared annotation. The integrated chat and the awareness features were positively mentioned six times. The negative comments were more diverse, and mostly addressed feature requests or small bugs. Besides the request for audio and video chat capabilities (5 times) the most reoccurring issues were: too verbose information in the chat (4 times), and too much vertical space occupied by the chat frame (2 times).

Conclusions and Outlook

Our work on SpreadVector informed the design of CoopFox, a tool that aims to provide lightweight support for collaborative Web research tasks, which is currently evaluated in a first user study. The design and evaluation of SpreadVector helped us to explore the interaction design space in combination with a technical lightweight implementation. CoopFox uses these foundations and added concepts concerning advanced issues like mechanisms for privacy, etc.

Acknowledgements

We thank the participants of the study and all members of the Cooperative Media Lab.

References

[1] Evans, B.M. and Chi, E.H. Towards a Model of Understanding Social Search. In Proc. 2008 ACM Conference on Computer-Supported Cooperative Work - CSCW 2008. pp. 485-494.

[2] Greenberg, S. and Roseman, M. GroupWeb: A WWW Browser as Real Time Groupware. In Companion Proc. Conference on Human Factors in Computing Systems - CHI 1996. pp. 271-272.

[3] Gross, T. Supporting Effortless Coordination: 25 Years of Awareness Research. Computer Supported Cooperative Work: The Journal of Collaborative Computing, Springer-Verlag 22, 4-6 (2013). pp. 425-474.

[4] Lowet, D. and Goergen, D. Co-Browsing Dynamic Web Pages. In Proc. 18th International Conference on World Wide Web - WWW 2009. pp. 941-950.

[5] Morris, M.R. A Survey of Collaborative Web Search Practices. In Proc. Conference on Human Factors in Computing Systems - CHI 2008. pp. 1657-1660.

[6] Morris, M.R. Collaborative Search Revisited. In Proc. 2013 ACM Conference on Computer-Supported Cooperative Work - CSCW 2013. pp. 1181-1192

[7] Perez, J.R., Leelanupab, T. and Jose, J.M. CoFox: A Synchronous Collaborative Browser. In Proceedings of 8th Asia Information Retrieval Societies Conference - AIRS 2012. pp. 262-274.

[8] Röscheisen, M., Mogensen, C. and Winograd, T. Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities. Computer Networks and ISDN Systems 27, 6 (Apr. 1995). pp. 739-749.

[9] Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Core (RFC 6120). http://www.ietf.org/rfc/rfc6120.txt, 2011. (Accessed 7/1/2014).