A Proxy-Based Infrastructure for Web Application Sharing and Remote Collaboration on Web Pages

Richard Atterer[1]
richard.atterer@ifi.lmu.de

Albrecht Schmidt[2]
albrecht.schmidt@acm.org

Monika Wnuk[1]
monius@fnuked.de

[1]Media Informatics Group
(Lehr- und Forschungseinheit
Medieninformatik)
University of Munich, Germany

[2]Fraunhofer IAIS, Sankt Augustin
B-IT, University of Bonn
Germany

A Proxy-Based Infrastructure for Web Application Sharing and Remote Collaboration on Web Pages

- Motivation
- Concept
- Implementation
- Evaluation
- Conclusion

Supporting Informal, Collaborative Use of the WWW

- Motivation
  - The web is regularly used collaboratively
  - Currently little support for remote collaboration – for example, must exchange links by text chat
- Proposed solution
  - Permit ad-hoc "sharing of the web browser", i.e. two remote users use a web application together
  - Rely on the existing browser platform for the implementation – use an AJAX-based approach
 - UsaProxy HTTP proxy, version 2

Concept of Web Application Sharing With UsaProxy

- Two users A and B reconfigure their browsers to use the web via UsaProxy 2

  - Remote monitoring: A sees B's actions, but cannot influence them, can only use built-in chat
  - Shared browsing: Symmetric operation, A and B can both interact with the site

- Aim: Non-invasive, ad-hoc operation possible, should work with most websites
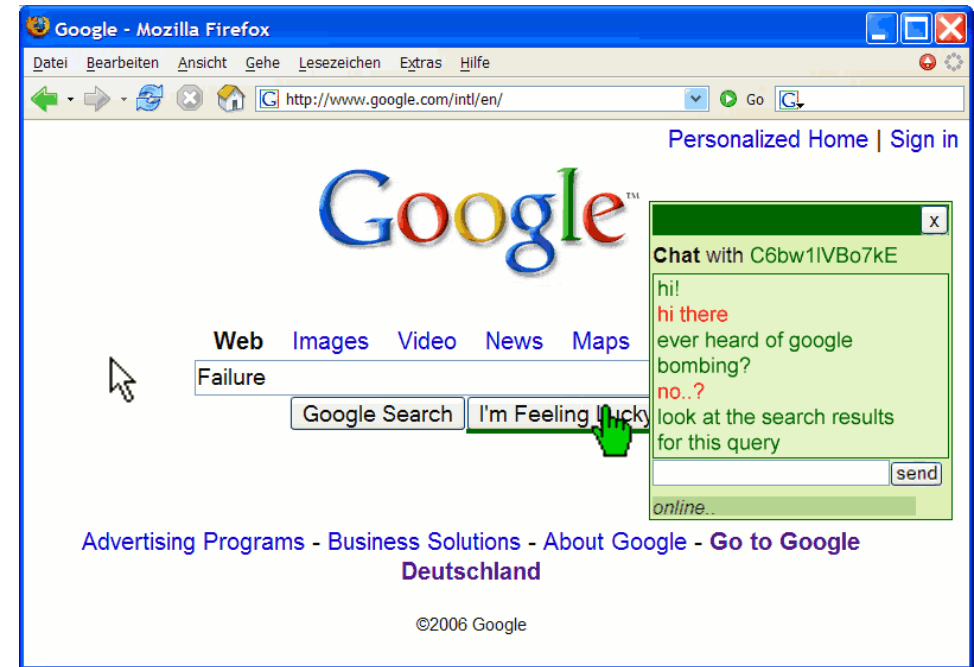
Concept of Web Application Sharing With UsaProxy

- Users share only one browser window/tab, the remaining ones are private. Inside shared window/tab:

  - Visualisation of remote mouse pointer movements and interaction (text selection, mouseover…)

  - Synchronisation of displayed content (including dynamic pages, following links, scrolling, simple AJAX applications)

  - text chat facility

- No true WYSIWIS (what you see is what I see) – rendering can differ due to window sizes, font settings…

- Shared session is started by visiting a special page

- Chat window indicates shared session – close it to stop sharing the session.

Motivation   Concept   Implementation   Evaluation   Conclusion
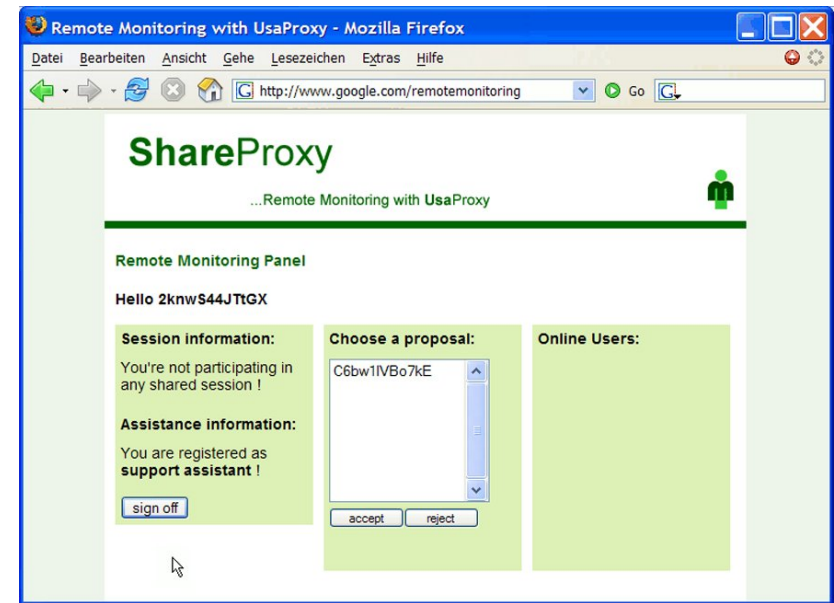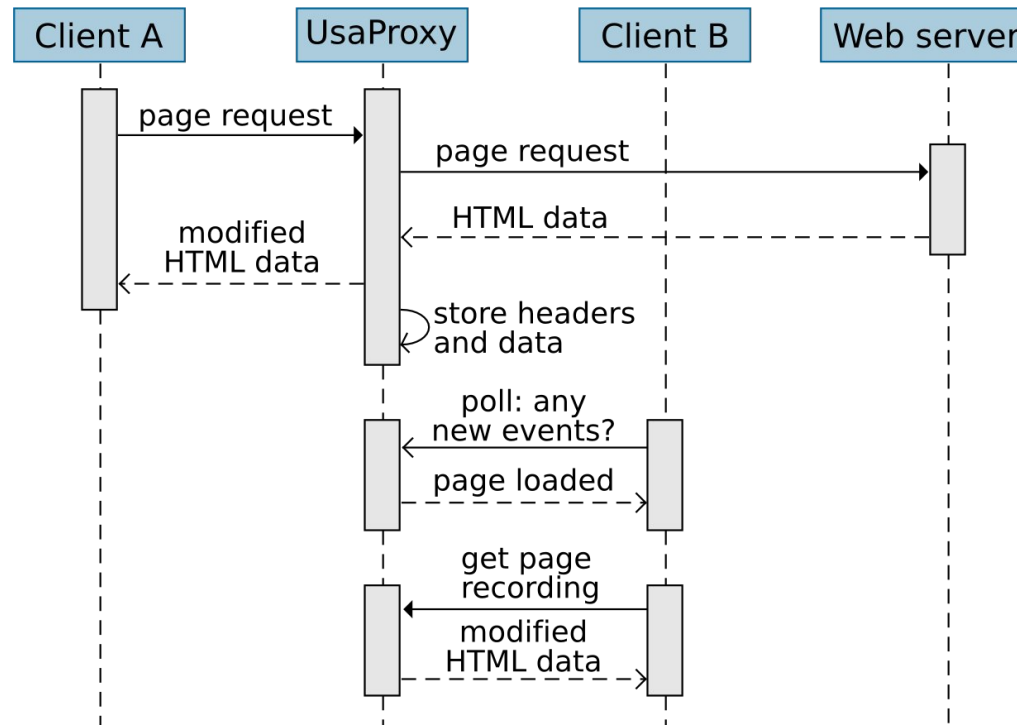
UsaProxy Implementation

- Open-source Java program, several modes:
  - Users manually reconfigure their browser
  - Transparent proxy for all users in a LAN
  - Transparent insertion in front of an existing server
- "text/html" server responses are modified before being passed on.
- The modification causes the proxy's JavaScript code to be loaded by the browser.
- The JavaScript code is executed on the client in the context of the page
- Log functionality

Client — Request → Proxy — Request → Server

Client ← modified HTML — Proxy ← text/html Response — Server
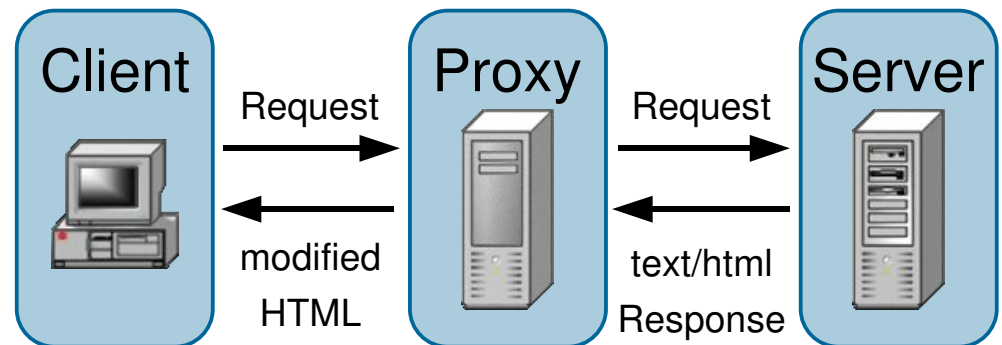
Replay of Pages in a Shared Session

- Single modification to the HTML code: Addition of
  <script src='http://84.2.100.71/usaproxylolo/file/proxyscript.js' type='text/javascript'></script>

- Request to fetch JavaScript and poll requests are intercepted, handled by UsaProxy behaving like a HTTP server

- Mouse movements etc. also transmitted as events

Motivation | Concept | Implementation | Evaluation | Conclusion

Implementation – Problems

- How to visualise mouse pointer despite rendering differences?
  - Identify hovered-over element by DOM path
  - place pointer relatively in its CSS box
  - Drawback: Pointer jumps possible
  - Problem: DOM tree differences MSIE/Firefox

- Need "global" (non site-specific) cookies to tell users apart on all websites.
  Trick: Set cookie for source site of UsaProxy JavaScript

- Clock differences between UsaProxy host and clients

- AJAX websites do not always work; both browsers contact the original server. Simple dynamic HTML works, as we record & play back *onmouseover* etc.

- User can type new URL in browser address line – we turn a window/tab permanently into "shared session tab" by assigning window.name = "sharedsession_UsaProxy";
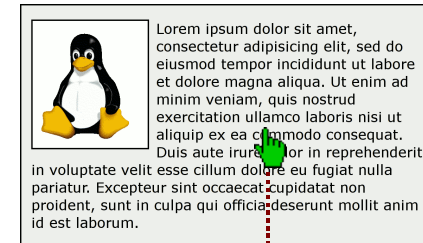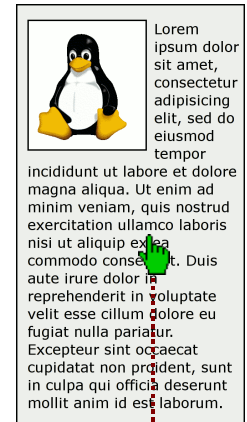
Evaluation

- Is the concept useful in practice to collaboratively solve tasks?
- Three different tasks
  - Both users are made to actively steer shared session
  - Remote user support: Website staff assists visitors
  - Teacher-learner scenario: Both take turns steering
- Small-scale test: 12 users (8 CS students, 4 staff)
  - Teams of 2 people
  - five-minute introduction to UsaProxy
  - Pre-configured test systems
  - Questionnaire after test
- Overall reactions positive

Evaluation Scenario 1: Looking for a Present Online

- Shared browsing mode

- *"On amazon.de, find a present to give to a common friend."*

- Problem: UsaProxy does not prevent conflicts, e.g. user A types in search field, B clicks on a link

- After initial confusion, social protocols developed

- Idea was liked despite problems

Evaluation Scenario 2:
User Support in a Web Shop

- Remote monitoring mode

- *"Complete the form on the website. Contact user support in case of problems."*

- Support user watched, gave advice via text chat

- Tailored towards server-side operation of UsaProxy: User does not need to reconfigure his browser

- 8 of 12 users would use this if offered on a site
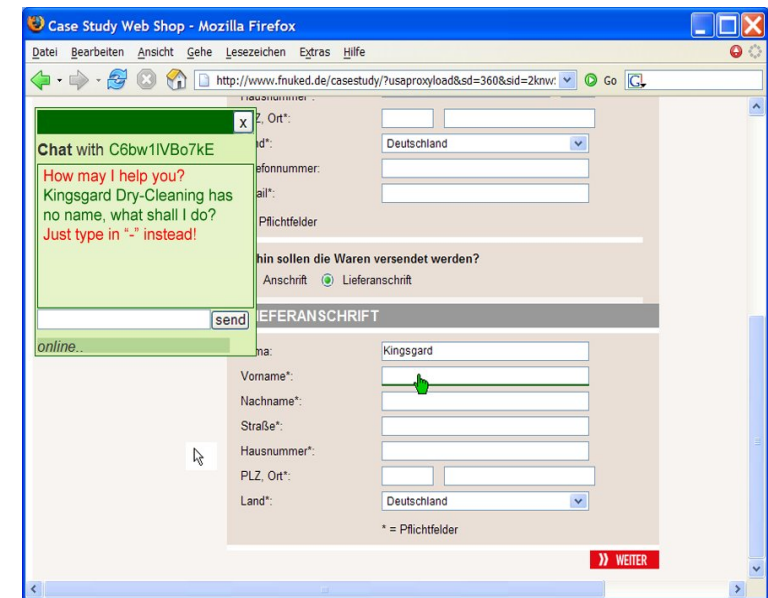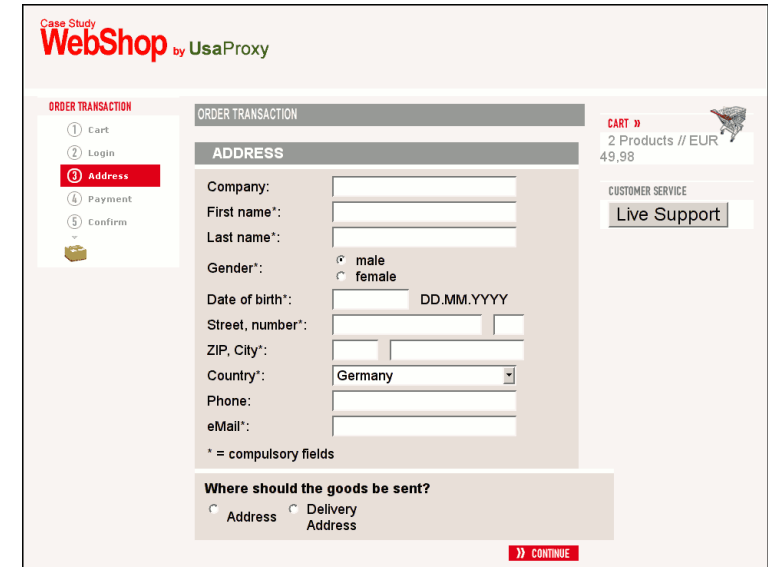
Evaluation Scenario 3:
Teaching the Use of a Web Application

- Shared browsing mode

- *"Search for certain information on the web. Let your friend help you if you cannot find it."*

- Our operator showed the user how to use Google advanced search

- Protocol implied: Handover of session via chat once user ran out of ideas

- Participants liked having seen all actions that led to the desired result
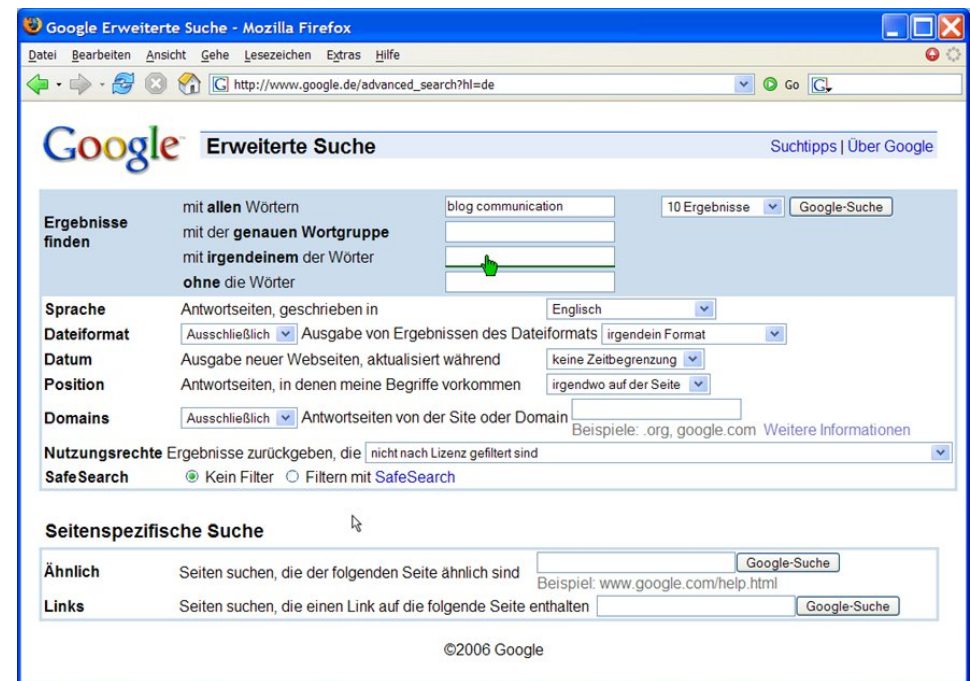
Discussion and Conclusion

- The technology works well, despite many challenges during implementation

- General technique (HTTP proxy + JavaScript) very useful for other applications

- Privacy concerns: Our solution does not allow "spying" on users, but the technology could be used for this – *only employ AJAX-based logging after users have explicitly agreed!*

- Security: Can direct browser to malicious websites, steal cookies… – *only use this with someone you trust!*

- Future work: For full support of AJAX applications, assymmetrical operation, copy DOM tree

- Special thanks to Monika Wnuk for great work on the implementation!

A Proxy-Based Infrastructure for Web Application Sharing and Remote Collaboration on Web Pages

Richard Atterer
richard.atterer@ifi.lmu.de

Albrecht Schmidt
albrecht.schmidt@acm.org

Monika Wnuk
monius@fnuked.de

UsaProxy 2 is open-source!

http://www.medien.ifi.lmu.de/team/richard.atterer/
http://fnuked.de/usaproxy/
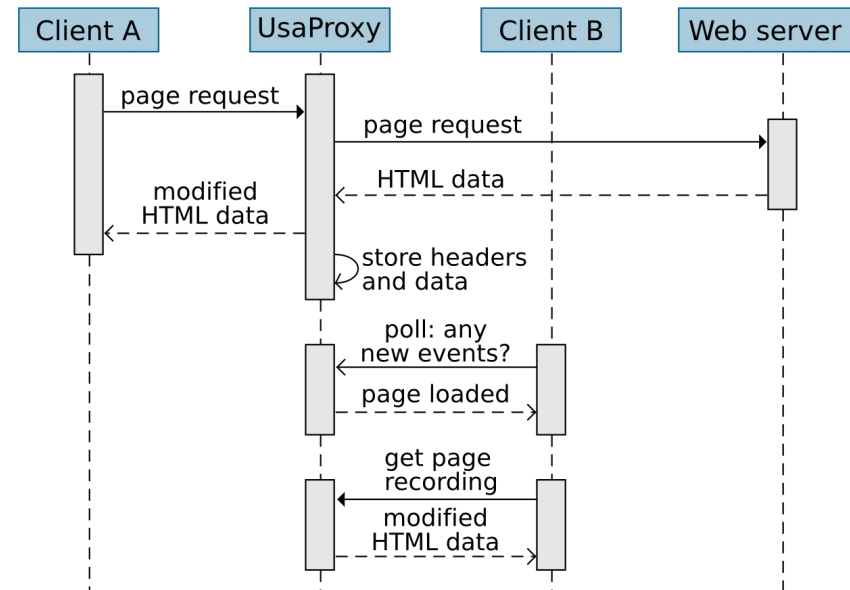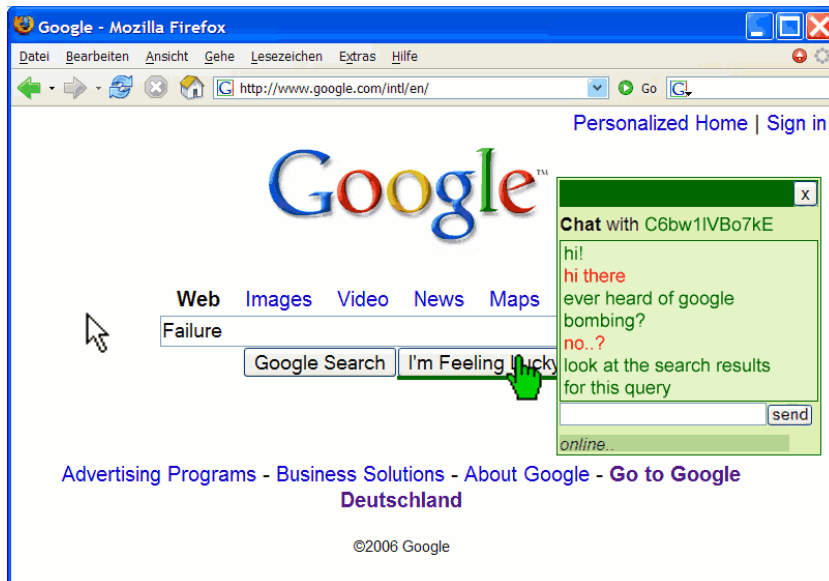
Comparison to Related Work

Distinctive features of UsaProxy:

- ## No sharing of the entire desktop, but only of one web browser window
  unlike VNC, remote desktop etc.

- ## No client-side software installation
  unlike GroupWeb [6], W4 [5], Kobayashi et al. [9]

- ## HTTP proxy approach
  as also used by Cabri et al. [3], WebSplitter [7]
  more robust than URL rewriting as used by CoBrow [11], CoWeb [8]

- ## Minimally invasive changes to pages
  no filtering of HTML code, unlike CoWeb[8], Cabri et al. [3]

- ## JavaScript-based
  also used by Esenther [4]
  unlike Java applets used by CoWeb [8], Cabri et al. [3]

Motivation   Concept   Implementation   Evaluation   Conclusion

References

(1) S. R. Ahuja, J. R. Ensor, S. E. Lucco: **A comparison of application sharing mechanisms in real-time desktop conferencing systems**. In ACM SIGOIS Bulletin, Volume 11, Issue 2-3, pages 238–248, April 1990

(2) R. Atterer, M. Wnuk, A. Schmidt: **Knowing the User's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction**. In Proceedings of the 15th International World Wide Web Conference (WWW2006), Edinburgh, Scotland, May 2006

(3) G. Cabri, L. Leonardi, F. Zambonelli: **Supporting Cooperative WWW Browsing: a Proxy-based Approach**. In Proceedings of the 7th Euromicro Workshop on Parallel and Distributed Processing, pages 138–145, Madeira (P), February 1999

(4) A.W. Esenther: **Instant Co-Browsing: Lightweight Real-time Collaborative Web Browsing**. In Proceedings of the 11th World Wide Web Conference (WWW2002), Honolulu, Hawaii, USA, 7-11 May 2002

(5) S. Gianoutsos, J. Grundy: **Collaborative work with the World Wide Web: Adding CSCW support to a Web Browser.** In Proceedings of the Oz-CSCW'96, DSTC Technical Workshop Series, pages 14–21, University of Queensland, Brisbane, Australia, August 1996

(6) S. Greenberg, M. Roseman: **GroupWeb: A WWW Browser as Real Time Groupware**. In CHI 1996 Short Papers: Proceedings of the Conference on Human Factors in Computing Systems, Vancouver, British Columbia, Canada, 13–18 April 1996

(7) R. Han, V. Perret, M. Naghshineh: **WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing**. In Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, United States, 2000

(8) S. Jacobs, M. Gebhardt, S. Kethers, W. Rzasa: **Filling HTML forms simultaneously: CoWeb – architecture and functionality**. In Proceedings of the 5th World Wide Web Conference (WWW1996), Paris, France, 1996

(9) M. Kobayashi, M. Shinozaki, T. Sakairi, M. Touma, S. Daijavad, C. Wolf: **Collaborative customer services using synchronous Web browser sharing**. In Proceedings of the 1998 ACM conference on Computer supported cooperative work, pages 99–109, Seattle, Washington, United States, 1998

(10) S. Puglia, R. Carter, R. Jain: **MultECommerce: a distributed architecture for collaborative shopping on the WWW.** In Proceedings of the 2nd ACM conference on Electronic commerce, pages 215–224, Minneapolis, Minnesota, United States, 2000

(11) G. Sidler, A. Scott, H.Wolf: **Collaborative Browsing in the World Wide Web**. In Proceedings of the 8th Joint European Networking Conference, Edinburgh, Scotland, May 1997

Motivation   Concept   Implementation   Evaluation   Conclusion