LiCoB: Lightweight Collaborative Browsing

Conference Paper · January 2009
DOI: 10.1109/WI-IAT.2009.352 · Source: DBLP

LiCoB: Lightweight Collaborative Browsing

Raphael O. Santos, Felipe F. Oliveira, Julio C. P. Antunes,
Magnos Martinello, Renata S. S. Guizzardi, Roberta L. Gomes
Federal University of Espírito Santo
{santos, ffrechiani, jcpantunes, magnos, rguizzardi, rgomes}@inf.ufes.br

Abstract

Collaborative navigation systems provide a useful way for virtual groups to share information through the web. However, the common set of features of these tools is not enough to offer a more face-to-face-like browsing experience. To fill this gap, this paper presents a novel collaborative navigation approach, which aims at integrating important features of a lightweight distributed architecture, awareness, session state sharing and annotations. In order to demonstrate the feasibility of our approach, LiCoB prototype was developed and then, evaluated considering performance issues.

1. Introduction

Due to its constant growth and evolution, the Internet has been considered a ubiquitous and important enabler for Computer Supported Cooperative Work (CSCW), favoring the sprouting of new collaboration services. Such services have introduced innovative interaction ways among users, providing better quality results in collaborative work, either for personal or professional purposes. Amongst these new forms of interaction, there is a paradigm known as co-browsing (Collaborative Browsing). This paradigm consists in allowing distant users to jointly navigate throughout Internet content, for instance, web pages.

Several application areas can take advantage of the collaborative Web browsing paradigm. For example, Web search, which is one of the most common online activities, is often undertaken in shared-computer context [1]. Educators have also verified the added benefit of co-browsing for teaching as it fits nicely into the theory of constructivism, allowing students to learn by exploring and sharing their own ideas and knowledge [2].

This paper introduces a co-browser named LiCoB that, besides supporting basic co-browsing functions, offers additional important functionalities, mainly related to a lightweight architecture and the maintenance of session awareness. More important, one of the main advantages of this approach concerns the introduction of annotation facilities, useful for expressing thoughts and, getting the focus of attendees on what the presenter wants to highlight. For example, while accessing online materials, annotations can help students to remember the key points of that lecture [3].

The remaining of this paper is structured according to the following organization: Section 2 presents some background information, besides a brief description of related work. Section 4 describes the system's design. Section 5 presents the LiCoB prototype and its main features. Section 6 shows some evaluation experiments focusing on system performance. Finally, section 7 concludes this paper, presenting some outlines of our future work.

2. Background and Related Work

Awareness in collaboration sessions is important for providing coordination and promoting usability [4]. Especially in a co-browsing session, it is important to be aware of the synchronization state of every participant. The user who is guiding the browsing activity of a group should be able to know if the other users' browsers have already loaded the shared content before making any comment about this content. Another important point about awareness is authorship awareness, which provides authoring information of users' participations during the sessions (e.g. users' annotations or browsing actions). Authorship awareness promotes user ideas contextualization, and can be applied to measure the importance level of the created information based on its source [5].

Several co-browsing systems do not provide any session history recording, like in [6] and [7]. Consequently, some collaboration information may be lost during session execution. This is an important drawback regarding, for instance, the support for session latecomers. It is known that Colab[8] and

Clavardon[9] keep a history of browsed URLs, while the system proposed by [7] creates a log of the users' action. However, logging actions and recording URLs may not be enough for allowing users to become contextualized with the session. For instance, when participants join an ongoing session, they may want to review previously presented information (e.g. URLs, annotations and coordination decisions).

Annotation feature is supported in Clavardon [9] and WebAn [10]. This is an important resource, allowing the participant to share information. However, the way it is implemented in such tools is not expressive enough since users are just allowed to highlight parts of texts. In LiCoB, annotations are allowed over all the shared content. This feature makes the collaborative browsing more powerful as it reduces the need for additional collaboration tools, enabling participants to share contextualized comments on each other's content.

Concerning system architecture, most co-browsers are based on a proxy server [8][6][9][11], that acts as an intermediate between websites and session members. The main advantage of this approach is that it does not require additional software installation on the client side. However, this architecture leads to potential performance problem, since the proxy server can become a bottleneck [8].

LiCoB relies on a server-based architecture driven by a lightweight synchronization protocol. This proposal is based on the distribution of session events which requires (i) lower bandwidth at the server side, since web pages contents do not pass through it; and (ii) lower processing capacity of the server, since the server handles only events. This approach avoids synchronization problems, since users are independent to download shared content directly from its web source. Therefore, users can leverage their own bandwidth, no longer depending on the connection to a specific proxy server.

3. Proposed Application Design

3.1. Synchronization Protocol

All the participations made by a leader follow a synchronization protocol in order to reach all attendees. This protocol is a specialization of the publish/subscribe communication paradigm [12], and in a nutshell, consists of small messages (usually less than 100 bytes) distributed to the participants of a session. Every message exchange is mediated by the LiCoB application server.

Illustrating the synchronization protocol, Figure 1 presents a common scenario which involves the broadcasting of a navigation (URL) from the navigation leader to the attendees. The main events composing this scenario can be explained as follows: (1) the leader notifies a new URL; (2) the leader starts to download the web page content; (3) the LiCoB server notifies all the attendees with the URL; (4) each attendee downloads the web page content by their own; (5) finishing the content download, the client informs the system LiCoB application server that it is synchronized; (6) finally, this server redirects such acknowledgment messages to the leader, keeping him aware of the attendees' synchronization state.

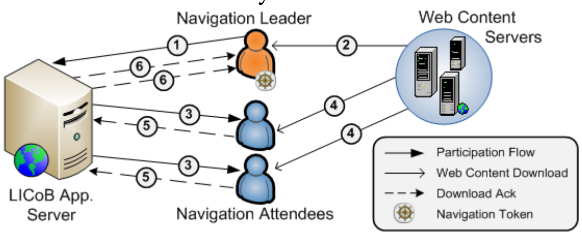

Figure 1. URL Broadcasting Scenario

The synchronization protocol provides a lightweight approach for LiCoB. Most of the data traffic is composed of small messages, not overloading the application server. This way, all the downloaded web content is exclusively dependent on the participant's network bandwidth, unlike proxy co-browsing solutions. Therefore, the LiCoB server does not become a navigation bottleneck for user downloads.

3.2. Architecture Components

LiCoB is designed to be a lightweight system composed by distributed components, i.e. a client and a server, as depicted in Figure 2. The application server acts as a synchronization point, dealing with a minimal workload to accomplish its task. Conversely, the client has greater responsibilities, saving server resources.

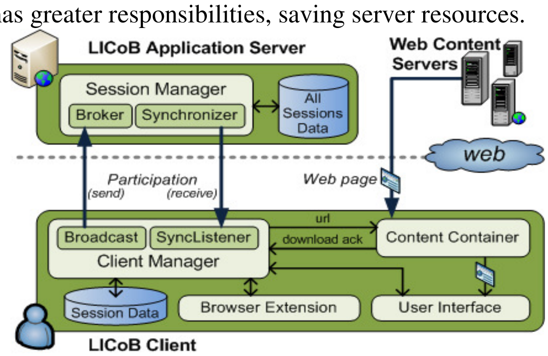

Figure 2. LiCoB Components

The server is essentially composed by the *Session Manager* module, which is responsible for handling all the received data. This manager is composed by two main communication sub-modules: on one hand, the *Broker* receives participations created on clients; on the other hand, the *Synchronizer* forwards them to the adequate attendees, elected by the *Manager* in accordance with the participants' receive privileges. In addition to that, the *Session Manager* uses the *Sessions' Data* repository to store all the information about the ongoing sessions.

In the same way, the *Client Manager* is the core of the LiCoB Client. Its main responsibilities are: observing user's actions, restricting the unprivileged ones; handling created and received participations; manipulating session awareness state using the Session's Data repository; and organizing user's interface events. Besides, the client is also composed by three other modules: (i) the *Content Container*, responsible for downloading web pages, acknowledging the system when finished; (ii) *Browser Extension*, designed to handle cookies and HTML interactions; and (iii) *User Interface*, in charge of presenting all the related information to the user and providing the system usage features.

4. LiCoB Prototype

The system's main navigation window (*User Interface*) is depicted in Figure 3, which illustrates an example of annotations painted over shared content (URL: http://www.ufes.br).

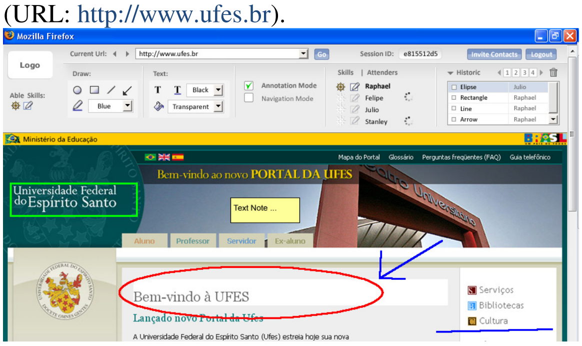

Figure 3. An Overview of the LiCoB User Interface.

On the top of the navigation window, we have the LiCoB Toolbar (Figure 4), which contains a set of interaction panels. The *Navigation Panel* allows the navigation leader to input the URL in which he wants to navigate. The *Annotation Panel* comprises the draw and comment tools (*ellipse, rectangle, line, arrow and text note*), besides and the controls to change annotation colors. By using them, one can point, mark, highlight, and comment passages of the presented content. A record of the participations is available in the *History Panel*, which also provides annotations authorship awareness. At last, *Users Panel* is responsible for showing the online participants and their current privileges depicted with token icons. Besides, the Users Panel allows the leader to pass the navigation token to another user. At last, it also shows a loading icon informing the Navigation Leader about everyone's synchronization state.

Aside from that, LiCoB supports session state sharing. This is an important feature that allows the collaboration session to share cookies created by web sites. Cookies are usually used for maintaining authenticated user's session and storing user's preferences. Thus, LiCoB allows its users to have the same view of the web site.

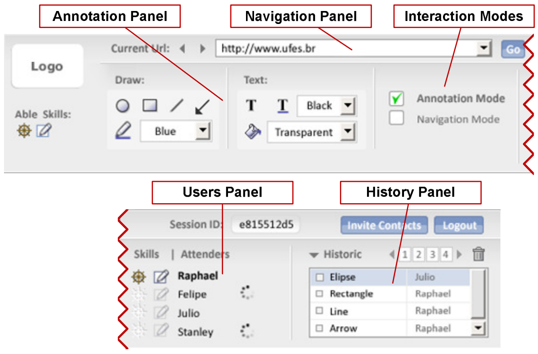

Figure 4. LiCoB Toolbar

5. Performance Analysis

This section describes a set of performance evaluation experiments using the developed prototype. The main objective here is to experimentally quantify *delay* imposed by LiCoB. In this regard, it is important to measure how long the system takes to provide a consistent synchronized session state for all participants.

Let the synchronization time be defined as the period of time taken by all the participants to reach the same session state. In other words, given a session, the synchronization time starts when its leader creates a Participation and finishes when this leader receives the last synchronization acknowledgement (see Figure 1 for more details). Finally, the *delay* is defined as the difference between the synchronization time and the web content download time. It represents the extra time required to keep the users synchronized, compared to single user browsing activity.

In order to quantify such *delay* measure, a set of experiments have been conducted, with real users (from 1 to 20) distributed over distinct Internet locations. Each experiment is composed of 10 web sites located at distinct domains (e.g. ".org" , ".com") with a content size varying from 500 to 4999 Kbytes.

Note that in the Internet scenario, the messages of the system are delayed by: (i) an inherent propagation (physical distance from participants to LiCoB server) (ii) a transmission delay (depending on users' bandwidth) and (iii) a network congestion (long queues at the routers). In this scenario, the users' Internet connection bandwidth varies from 100Kbps to 10Mbps. Additionally, the user's network path to the LiCoB server has around 15±10 hops, with an average delay of 230±150 ms. Figure 5 depicts the results for this scenario, in which the *delay* had an average value of 559±163 ms.

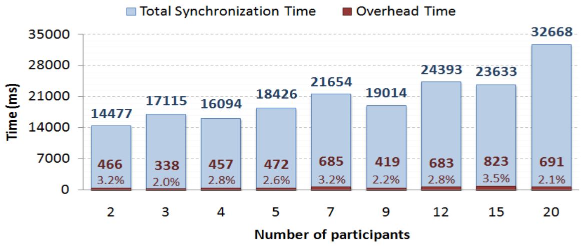

Figure 5. Internet Experiments

The traditional navigation (single mode) paradigm is naturally affected by the conditions of a heterogeneous environment. Due to the design of LiCoB's architecture, users perform web content download as if they were in single mode, thus being identically affected. The only difference between LiCoB's navigation and single navigation is given by the *delay*. However, as the results discussed above have shown, this time is short enough keeping its value around 2.7% ± 0.5% of the web page download time.

6. Conclusion and Future Work

This work presented a co-browser called LiCoB, created to handle web content sharing, annotations and awareness. The system is based on a signalization broadcasting protocol developed to keep users synchronized while avoiding network overload. This protocol helps maintaining a lightweight system performance.

Comparing with the most adopted proxy-based approach, LiCoB's architecture avoids bottlenecks on the server-side, taking advantage of each participant's bandwidth, in order to promote a better synchronous collaborative experience. Preliminary evaluation experiments showed satisfactory quantitative results on session synchronization. In the future, we intend to improve the evaluation procedure by including larger experiments and some qualitative methods in order to evaluate cooperation aspects and system usability.

Even more, the LiCoB conceptual model was built based on a Collaboration Ontology [13], which saved time and improved quality in the system's development process.

7. References

[1] Amershi, S., Morris, M.R. "CoSearch: A System for Co-located Collaborative Web Search" *CHI 2008, ACM*, New York, NY, April 5–10, Florence, Italy, 2008, pp.1647-1656.

[2] Aneiros, M., Estivill-Castro, V., "Usability of Real-time Unconstrained WWW-co-browsing for Educational Settings", *Proceedings of the 2005 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI'05),* Sept 2005, pp.105 – 111.

[3] Chong, N.S.T., Sakauchi, M "Creating and Sharing Web Notes via a Standard Browser", *Proc. of the SAC*, Las Vegas, ACM, New York, NY, *2001,* pp. 99-104.

[4] Gutwin, C. and Greenberg, S. "Effects of awareness support on groupware usability". In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'01).* ACM, New York, NY, 2001, pp.511-518.

[5] Schmidt, K. and Bannon, L., "Taking CSCW Seriously: Supporting Articulation Work", *CSCW: An Int.l Journal*, 1992, pp. 1(1-2):7-40.

[6] "Cobrowser.net". [online: http://www.cobrowser.net].

[7] L. Gerosa, A. Giordani,, M. Ronchetti, A. Soller,, R. Stevens, "Symmetric Synchronous Collaborative Navigation", *Int. Conf. WWW/Internet*, 2004, pp. 748-754.

[8] G.J. Hoyos-Rivera, R.L. Gomes, R.C. Willrich, J.P., "CoLab A New Paradigm and Tool for Browsing Collaboratively the Web", *IEEE Transactions on Systems, Man, and Cybernetics*, v.36, n.6, 2006, pp. 1074-1085.

[9] "Clavardon" [online: http://www.clavardon.com].

[10] C.C. Marshall, and A.B. Brush, "Exploring the relationship between personal and public annotations". *In Proc. of the 4th ACM/IEEE-CS Joint Conf. on Digital Libraries*, Tuscon, AZ, USA, June 2004, pp. 349-357.

[11] K. Maly, M. Zubair, L. Li, "CoBrowser: Surfing the Web Using a Standard Browse", *World Conf. on Educational Multimedia, Hypermedia and Telecommunications*, 2001.

[12] P.T. Eugster, P.A. Felber, R. Guerraoui and A. Kermarred, "The many faces of publish/subscribe". *ACM Computing Surveys*, ACM, New York, NY, USA, v. 35, n. 2, p. 114–131, June 2003.

[13] Oliveira, F.F., Antunes, J.C.P. and Guizzardi, R.S.S., "Towards a Collaboration Ontology", *in Proc. of the Snd Brazilian Workshop on Ontologies and Metamodels for Software and Data Engineering*, João Pessoa, Brazil, 2007.