A generic, distributed Whiteboard overlay using Java and the MBone

Mattias Mattsson, Peter Parnes, Kåre Synnes, Dick Schefström
Department of Computer Science/Centre for Distance-spanning Technology
Luleå University, Sweden
{Mattias.Mattsson,Peter.Parnes,Kare.Synnes,Dick.Schefstrom}@cdt.luth.se

Abstract

This paper presents the idea of a shared whiteboard that can be used as a generic overlay over different applications. Thus we would obtain an annotation tool for different applications, such as document viewers and WWW-browsers, as well as a conference tool where several members can annotate a document during discussion.

1 Introduction

Today the World-Wide Web (WWW) [13] is a read-only media. The WWW-pages only shows what their authors wrote and it is often impossible to make annotations on WWW-pages, i.e. links to related pages or comments made by readers. When it is possible, the annotations usually have to be on a separate page. It would be an advantage if these annotations would be graphics that are drawn on top of the WWW-page and not just text-only annotations at a separate page.

Another aspect of the WWW is that it is for single user mode only. We cannot gather a set of persons around a WWW-page and draw real-time graphics on the WWW-page while discussing its topic. If we could do so, and the WWW-browsers of the participants were synchronized so that when you changed page, the others would also have that page loaded automatically, then this could be used for a WWW-conferencing system. With some conference control support, such a system could be used for conferences ranging from a few persons discussing a WWW-page to a distributed lesson with hundreds of students, where the teacher controlled the students WWW-browsers. It should also be possible to save those lessons to disk along with time-stamps of the events so it could be re-played later.

These two functions, annotation and conferencing, share the need for drawing graphics, a whiteboard tool. Such a whiteboard tool could be used outside the WWW as well, even though the conferencing aspect is the most obvious then. A disadvantage of having the whiteboard tool tightly coupled with a specific application is that it would restrict annotation and conferencing to that application, or even restrict it to either annotation or conferencing, whereas having a generic whiteboard tool that was combinable with an arbitrary application would give the same facilities to any application. Furthermore, since the conferencing part of the whiteboard tool requires scalable, reliable communication among the participants, a feature well outside the ordinary WWW-browser, it will probably not be incorporated in any WWW-browser in the near future.

1.1 Related work

Work have been done previously in this field, the NCSA Mosaic [5] browser allows you to annotate a document but these annotations are only a link at the top or the bottom of the document to a separate page with the actual annotation. The annotation is text-only, although you can HTML format it. It is impossible to do anything else than personal annotations which a disadvantage when you make annotations that would be of interest to a group of people.

The ComMentor [1] system, which is a separate extension of the NCSA XMosaic browser, allows you to add annotations in-line in the WWW-page. The annotations show up as bullets in the text and you can get a preview of the annotation by clicking on it with the middle mouse button or see the whole annotation by selecting the bullet with the right mouse button. The annotations are also here text-only and on a separate page. An effect of that the annotation bullets must be in-lined in the text is that it might be quite hard to read WWW-pages with many annotations. A graphical annotation system would probably be more readable because even if it *allows* users to annotate on top of the text, it does not *force* them to do so. The ComMentor system has solved the problem of providing the annotations to other persons by having annotation servers and annotation sets so that when you go to a WWW-page, the WWW-browser asks the annotation server if there are any annotations on that page in the annotation sets that you are a member of.

When it comes to conferencing systems, there exist tools, e.g. the wb [10] system, that allow users to graphically annotate documents using scalable, reliable communication but these are usually restricted to post-script or text documents.

1.2 A whiteboard tool

A whiteboard tool would benefit from being developed as a separate part and not tightly coupled with an application but in order to be useful for these purposes, this whiteboard tool would have to fulfill certain demands:

1. It should of course supply all the drawing primitives, e.g. lines, circles, free-hand drawing, that is needed for annotations but also other features that are necessary for a conferencing system.

2. It should supply means for saving and loading drawings along with time-stamps to be able to play recorded sessions at a later time. The saving and loading should be generic enough to allow connections with annotation servers if such are available.

3. It should have scalable, reliable and real-time, communication with all participants (in the conferencing scenario) so that you can see not only what previously have been drawn on a WWW-page, but also what other participants draw at the moment.

4. Last, but most important, it must be easy to combine the whiteboard tool with an arbitrary application with a minimum of work since this whiteboard tool is not very useful on its own.

Apart from the whiteboard tool there are other utilities that has to be available if both annotations and conferencing shall be possible. The conferencing part demands that the participants at least have an audio-conferencing session so they can talk about the WWW-page or document that they are viewing. Annotation of WWW-pages would demand a separate annotation server so that other persons can see the annotations you have made. This is outside the scope of this paper but should not be too difficult to solve, audio-conferencing tools such as VAT [9] already exist and private annotations, at least, would not be more difficult than saving the annotations to disk.

2 WebDesk

Work is being made on a distributed conference environment, WebDesk [11], here at CDT, which is a part of the MATES project. WebDesk shall allow the users to interact with each other through audio, video, drawing and annotation of existing documents. It is being developed in Sun Microsystems new Java [3] technology.

One of the parts implemented in WebDesk so far is an implementation of a reliable multicast framework called Scalable Reliable Real-time Transport Protocol (SRRTP) [8]. SRRTP is an extension of the Real-time Transfer Protocol (RTP) [6] that incorporates the Scalable Reliable Multicast (SRM) [7] framework. SRRTP is, since it is an extension of RTP, independent of the underlying network but the implementation used in WebDesk is based on the Multicast Backbone (MBone) [4]. This allows scalable, reliable and real-time, communication with all participants in a conference.

WebDesk also has an early implementation of a whiteboard tool [12]. This whiteboard allows users to draw graphic primitives, e.g., lines, circles, free-hand drawing, writing text. The whiteboard works with the concept of documents which each is an ongoing conference. Each document contains a number of pages, which may be a page in a post-script document or a WWW-page. Each page may in turn consist of several layers so that each participant can get her own layer or even several layers. These layers can be thought of as transparent slides that may be put on top of each other to form a page. This allows a more fine-grained control of the conference since you can turn off one or more layers to exclude drawings made by certain members instead of just being able to chose to either view the drawings or not. Furthermore, this allow users to prepare some comments on a separate layer and, when finished, present them to the rest of the group.

Other features are tele-pointers that allow you to attract other members attention to certain areas. In wb you had to draw something to do this which is not as convenient since you might not want to leave marks everywhere on the document. There are also radarviews which are small windows that show a miniature version of a certain page in the whiteboard document. This allow you to have an overview of several pages at once while working on another page.

The drawing primitives in this whiteboard are separate Java classes that knows how to construct, paint, and transform themselves into and out of bytes. This means that the whiteboard does not have to know anything about the primitives other than that there is a set of methods that all primitives ensures is implemented. Hence it is possible to add new primitives without changing the whiteboard at all. This may even be done during run-time.

Since these primitives know how to transform themselves into and out of bytes, which is used when sending them to other members of the conference, they may also be saved to disk or at an annotation server by redirecting the stream of bytes accordingly.

So far, the WebDesk environment solves the first three demands we had on a whiteboard tool but the fourth, and most important demand still remains, it should be easy to combine the tool with an arbitrary application. Due to the facts that WebDesk is written in Java and that all graphical Java components descends from a common ancestor class, it is possible to change other Java applications so that they inherit the whiteboard drawing surface component, rather than the standard Java graphic component. This works since the whiteboard drawing surface ultimately is a standard Java graphic component itself. It has to be done this way since Java does not support multiple inheritance. By changing a small amount of code in the original application, depending on the desired level of integration, it is possible to have the whiteboard tool to show its graphics on top of the original Java application.

This way of solving the integration narrows arbitrary application to arbitrary Java application with source code available, but is still a substantially gain from whiteboard tools tied to one program. This approach is easier than using shared-X applications and also allows the whiteboard tool to work on all Java supported platforms without extra effort. There already exist a WWW-browser that is written in Java, the HotJava [2] browser, and hopefully post-script viewers and other document format viewers will be developed soon.

Further work

Most important is to get access of the HotJava source code and do an integration between this browser and the whiteboard tool offered by WebDesk. This is the crucial part of the whole idea. Although this has not been tested yet, some tests of this concept has been made and the whiteboard tool has been integrated with a stand-alone, albeit simple, Java application that shows images and thereby achieving a conferencing tool for images. Even if this does not ensure that integration with larger applications will work, it shows that the concept is possible.

Another major problem when integrating the whiteboard tool with HotJava is to tie the graphical annotations together with a WWW-page. A WWW-page, unlike a post-script document, may change totally when it is resized or when you change some of its properties, e.g. font style or size, thereby making annotations that are fixed to a certain coordinate invalid. One solution, although not very elegant, would be to fix the size and properties of the browser window so that the layout of the WWW-page is the same and the annotations make sense. Another, more elegant solution, would be to integrate the browser and the whiteboard tool to cooperate and tie the annotations to the semantic context of the WWW-page. This would be possible if the graphic primitives of the whiteboard were aware of the WWW-pages semantics since they are responsible for drawing themselves on an area. Thus, if the primitives were given a reference to a Java WWW-page object, they would be able to draw themselves in the right place given a certain context. This could be accomplished by only changing the drawing method in the drawing primitives.

3 Conclusions

This paper has described the advantages of having a distributed whiteboard tool that can be used as a generic overlay over other applications. Such a tool should have support for ordinary drawing primitives, loading and saving of the drawings along with their time stamps, have support for scalable, reliable and real-time, communication, and finally be easy to integrate with existing applications. This gives a whiteboard tool that, with some external help, is suitable for both annotation and conferencing around chosen media, may it be WWW or something else.

By doing the whiteboard tool in Java it is possible to integrate the tool with other application with small modifications of the existing code.

Acknowledgments

This work was done within Esprit project 20598 MATES, which is supported by the Information technology part of the 4:th Framework Program of the European Union. Support was also provided by the Centre for Distance-spanning Technology (CDT).

References

[1] M. Röscheisen, C. Mogensen, T. Winograd, *Shared Web Annotations As A Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples*, <URL: http://www-diglib.stanford.edu/diglib/pub/reports/commentor.html>

[2] JavaSoft Inc., *HotJava*, <URL: http://www.javasoft.com/>

[3] JavaSoft Inc., *Java*, Technical Report, 1996, <URL: http://www.javasoft.com/>

[4] *The MBone information web*, <URL:http:// www.mbone.com/>

[5] *NCSA Mosaic home page*, <URL: http://www .ncsa.uiuc.edu/SDG/Software/ XMosaic/>

[6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications* - IETF RFC1889, January 1996.

[7] S. Floyd, V. Jacobson, S. McCanne, C. Liu, L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Framing", *ACM SIGCOMM'95*

[8] P. Parnes, *Scalable Reliable Real-time Transport Protocol - SRRTP* , work in progress, <URL: http://www.cdt.luth.se/˜peppar/docs/>

[9] V. Jacobsen and S. McCanne, *vat - X11-based audio teleconferencing tool*, UNIX manual page, Lawrence Berkeley Laboratory, University of California, Berkeley.

[10] V. Jacobsen, *A Portable, Public Domain Network 'Whiteboard'*, Xerox PARC, viewgraphs, April 1992.

[11] P. Parnes, D. Schefström, K. Synnes, *Web-Desk: the Collaboration Support in MATES*, First European Workshop on Global Engineering Network, 1996, <URL:http://www. cdt.luth.se/˜peppar/docs/>

[12] M. Mattsson, *Whiteboard*, work in progress, <URL: http://www.cdt.luth.se/˜mattias/work/>

[13] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Nielsen, A. Secret, "The World-Wide Web", *Communications of the ACM*, August 1994