Registering Web-Based Conferencing With Structured XML Documents

Maria da Graça Pimentel, Marcos A. S. Kutova
Instituto de Ciências Matemáticas e Computação - Universidade de São Paulo – Brazil
mgp@icmc.sc.usp.br, kutova@icmc.sc.usp.br

Cesar A.C. Texeira
Departamento de Informática – UNIFACS – Brazil
cesar@dc.ufscar.br

Abstract: The aim of the work reported in this paper is to provide an environment where users exploit the Web as a platform to communicate using simple conferencing tools at the same time that have the memory of that communication registered as structured hyperdocuments for later retrieval an reuse.. We present DocConf, an extensible tool to support Web-based conferencing that provides the registering of the communication occurring in a conferencing session as an XML-structured hyperdocument, as well as relies on such hyperdocument to present all the previous activities of a session to newcomer users.

1. Introduction

Douglas Engelbart, one of the visionaries of computer science and technology, was among the first to discuss the advantages of exploiting structured hyperdocuments to provide for interoperability between CSCW tools. His proposal, the Open Hyperdocument System (OHS), would provide standards and facilities to the creation, storage, retrieval and interchange of knowledge by using hyperdocuments [Engelbart 90]. Elgerbart's recent view is that the World Wide Web (Web) can expand to provide a platform for the goal of augmenting communities of collaborating knowledge workers. His suggestion is that the Web grows towards to become the OHS, which would demand, among others, integrated object oriented applications, explicitly structured documents and view control of form, sequence and content [Engelbart 95, 98].

Commenting on the benefits resulting from combining the technologies of hypermedia and computer supported collaborative work in the early 90's, Streitz suggested that such a combination would result in multi-user distributed hypermedia systems at the same time that CSCW tools could benefit from the use of structured hypermedia hyperdocuments. In the same context, Ishii specifically suggested the use of structured documents to register the memory of the group in CSCW applications [Streitz 91].

There are many efforts being made in terms of allowing for the specification of the structure of the documents in a standardized way, a case in point is the XML specification recommended by the World Wide Web Consortium (W3C), as discussed in Session 4.

Moreover, recent work has been made in order to exploit the Internet as an environment for synchronous communication in order to support collaborative work. An important reference is Habanero, a software suite designed to facilitate the development and integration of collaborative Internet applications [Chabert 98].

We have been working in the development of an extensible software suite that provides infrastructure for synchronous and asynchronous collaborative work over the Internet in general, and the Web in particular. In this project, we investigate flexible ways to support the automatic generation, storage, retrieval and presentation of the information generated in a collaborative working session. In this context, we have build DocConf, an extensible tool to support Web-based conferencing that provides the registering of the communication occurring in a conferencing session as an XML-structured hyperdocument, as well as relies on such hyperdocument to present all the previous activities of a session to newcomer users.

The aim of the work reported here is to provide an environment where users exploit the Web as a platform to communicate using simple conferencing tools at the same time that have the memory of that communication registered as structured hyperdocuments for later retrieval and reuse. An important requirement is that the document generated be structured according to an open recommendation to allow information interchange and manipulation.

The basic conferencing services provided include a chat tool for exchanging typed messages, a common whiteboard and support to synchronous voting. The environment is easily extensible to facilitate the integration of tools related to other media (such as audio and video) or tasks (e.g. shared Web browser).

The remaining sections of this paper are organized as follows. The next section states the requirements defined

for DocConf, as well as presents information on its current stage of implementation. The following section details aspects of the architecture of communication in DocConf. Next, the advantages of using XML to formalize the structure of the generated documents are listed, preceding a section that details the XML specification for a hyperdocument of a DocConf session. Following, a section gives an overview of typical steps taken by a user when entering a session, relating them with the generation of the underlying hyperdocument. Sections on related work and final remarks conclude the paper.

2. Requirements and Current Stage

This section lists the functional requirements specified to DocConf, as well as describes how the functionality is delivered, if implemented, or the current stage of its implementation, if under development.

Support to synchronous sessions. A session begins when created by the first participant (the owner) and ends when the last participant leaves: the server module of DocConf keeps track of logins and all other requests.

Registering the exchanged communication in a structured hyperdocument. XML was chosen as the specification language (Section 4), and a specification named DocConf DTD was defined to support a DocConf session (Section 5). The DocConf server registers all messages exchanged among the clients as the corresponding XML elements.

Support communication by a chat tool, a common whiteboard and a voting tool. Three tools where implemented and can be loaded as part of a session or not, depending on the configuration given when the creation of the session. However, a tool not requested at the time of the creation of a session can be loaded at any time during the session.

Presence reporting. DocConf may be set to indicate all the users present in any session at a given time. This type of information is important to support small and large groups [Benford 97].

Allow late login: when a user enters an open session, the conferencing tools (chat, whiteboard and vote) show the information previously discussed: the client module of DocConf requests from the server the information XML document corresponding to the session.

Support to asynchronous sessions. Support in DocConf to asynchronous sessions is currently under development, and exploits the component-based architecture of the current implementation (Section 3).

3. Architecture of Communication

DocConf is being implemented in Java: Swing is used for the implementation of the graphical user interface, and sockets TCP/IP are used to provide fast and reliable client/server communication.

DocConf uses an architecture of communication based on components, which correspond to the rectangles illustrated in [Fig.1]. A "Session Server" module has one component for each session it controls ("Session Alfa" in "Session Server" of [Fig.1), which in turn has one component has one component to each user logged in the session ("ClientThread" in the "Session Server" of [Fig.1]).

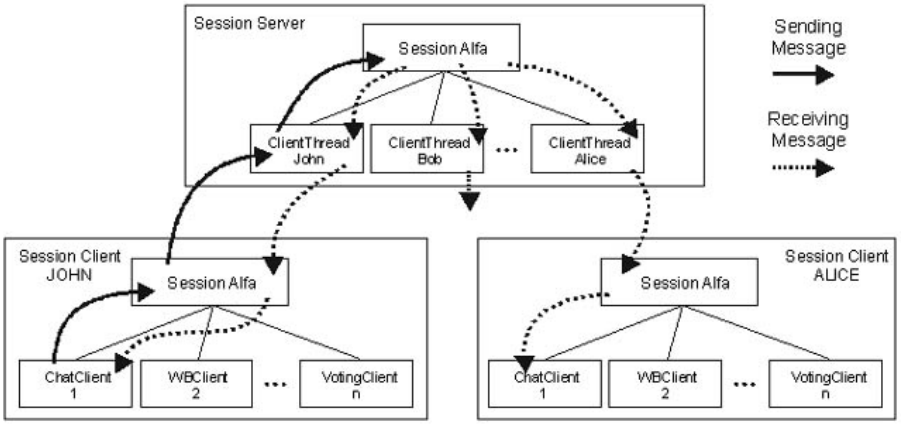

Figure 1: The flow of information between DocConf client and server modules

Similarly, each "Session Client" module has one component for the session it corresponds to ("Session Alfa" in the "Session Client" of [Fig.1]), as well as one component for each tool used in the session (e.g.: "VotingClient").

The functionalities of DocConf are implemented using components, each component having two objects:

interface and message. Messages are objects generated by the interface and send to/received from the server. A server forwards the information received without processing it: specific processing is done on the interfaces in the client modules (as indicated in the flow of messages in [Fig.1]).

The interface is responsible by interacting with the user, communicating messages and implementing the functionality of the tool (e.g.: drawing in the whiteboard) with the server. Messages correspond to all data exchanged among modules, including control information (e.g.: user login in) and content (e.g.: user identification).

A user can participate in more than one session simultaneously, and several types of objects manage the underlying communication. As illustrated in [Fig.2], a user runs one instance of the object "Client", which owns one object "SessionClient" for each session the user is in. Similarly, the object "Session Server" owns one object "Session" for each session managed. A "Session" object creates a "ThreadClient" object for each user (a "SessionClient" object), communicating in that session.

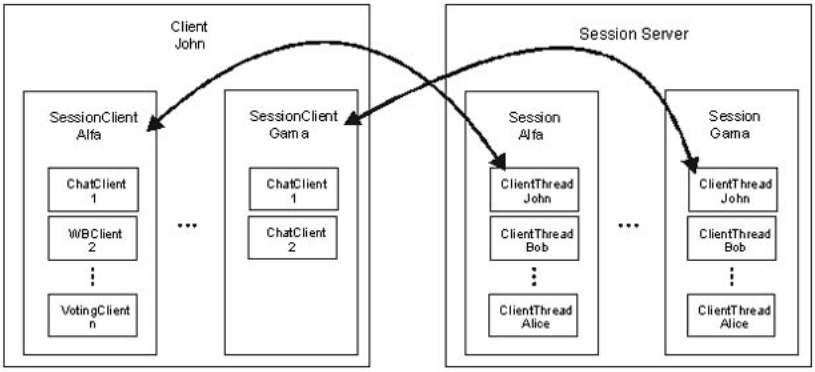

Figure 2: Class hierarchy for DocConf client and server modules

4. Advantages of Using XML

Referring to the work of the Word Wide Web Consortium, Rada et al. state that standards are the body and soul of the Web [Rada 98]. The World Wide Web Consortium (W3C) has been investing many efforts in order to guarantee the growth of the Web in an open and standardized way, for instance by defining recommendations regulating the formalization of the structure of documents and their presentation format. Specifically, the Extensible Markup Language (XML) helps domain application designers to provide a logical structure for the supported documents [XML 98], while Cascade Style Sheets (CSS) and Extensible Style Sheet Language (XSL) help those authors to standardize the presentation format and reuse of their document [CSS 96, XSL 99].

Although XML aims at supporting intra-domain interoperability, there is still the problem of providing distinct domains a way to cross communicate. To solve the latter, the W3C has just released the specification of the Resource Description Framework (RDF), a foundation for processing metadata and make information exchangeable across application domains on the Web.

The structuring and storage of the information produced during a session using a standard markup language such as XML bring the advantages described below, some of them common to any XML based-application as discussed in [Bosak 98, Connoly 98, Glushko 98]:

Interoperability among applications. It is not necessary for applications, mainly third party ones, to know details of a particular database structure, which could be used as an alternative to XML for internal storage purposes.

Distributed processing. Besides third party authorized applications, which may require access to the XML document produced during a session, all the clients, representing the participants of the session, are continuously receiving from the server the events of the session. Hence they are able to keep updated their local version of the XML document, which reflects the state of the session until the considered moment, and to promote local processing in order to recover and manipulate particular information.

Customized view. As a result of the flexibility given by the XML, the clients can customize the presentation of the partial document during the session, or of its totality, afterwards. That is quite handy to help and support their participation in the cooperative work and for posterior analysis of the session. As an example, a participant of a meeting would like to keep in her screen, during the session, a particular view referring to her interventions.

Semantic attribution. For the more structured or formal sessions, maybe for the asynchronous ones, or for those related to a specific area, participants may find useful to contribute with marked up contents in order to allow

semantic attribution to the information and facilitate its automatic processing. In such a situation, structuring all the session as a XML document is likely to be a good and homogeneous approach.

Independent searching. Taking a chunk of information as a XML document allows any application to browse freely the document and to extract the information needed.

Reduced traffic. The possibility of browsing the document locally, at the client side, may be more efficient, in terms of network traffic, for applications that would instead require several queries to a remote database.

Fast searching. Another advantage of browsing the document locally is speeding searching procedures.

5. The DocConf Specification

The DocConf Document Type Definition (DTD) shown in Figure 3 is the specification for the documents generated in the DocConf sessions. Each document is composed of the elements <head> and <body>. The <head> element stores information relative to the creation of a session (<name> and <moderator>). The <body> element contains control and content information corresponding to the communication that occurs among the users, and as such contain the elements <login>, <logout>, <chat>, <whiteboard> and <vote>.

Within the <body> element, many elements have other control information specified in terms of attributes of the corresponding element, such as identification information <compid> or a timestamp <date>.

The element <cstart> stores information about the initialization of a new component. It may have some <param> elements to hold extra information like component layout or initializing options.

The <chat> element contains information manipulated by the chat tool, and has elements <message> and <from> for each message exchanged. The element <whiteboard>, in turn, keeps information manipulated by the whiteboard tool, and has elements such as <shape> for empty shape, <filledshape> for filled shape, <text> for typed text, and <polyline> for polygons. Among the attributes of these elements are <type> for specifying whether a shape is a circle for example, <coords> for position of the element on the whiteboard, <foreground> for the color of the object, and <background> for the filling of a filled shape object. The element <vote> keeps the votes for voting situations. The voting options are defined in the parameters of the <cstart> element.

```
<!ELEMENT  session     (head,body)          >       <!-- Chat tool elements                  -->
<!ATTLIST  session  date  CDATA  #required >       <!ELEMENT  chat        (from,message)      >
<!ELEMENT  head        (id,moderator)       >       <!ELEMENT  from        (#PCDATA)*          >
<!ELEMENT  id          (#PCDATA)*           >       <!ELEMENT  message     (#PCDATA)*          >
<!ELEMENT  moderator   (#PCDATA)*           >       <!ATTLIST  chat compid CDATA  #required
<!ELEMENT  body        (login | chat |                           date   CDATA  #required    >
    whiteboard | voting  | vote | logout)* >       …
<!ELEMENT  login       (#PCDATA)*           >       <!-- Voting tool elements                -->
<!ATTLIST  login    date  CDATA  #required >       <!ELEMENT  vote        (#PCDATA)*          >
<!ELEMENT  logout      (#PCDATA)*           >       …
<!ATTLIST  logout   date  CDATA  #required >
```

Figure 3: Portion of the DocConf Document Type Specification

6. DocConf Interaction and Document Creation

The typical steps taken by a user to participate in a session are presented to illustrate how the associated structured document is created. In each case, the elements for the DTD involved in the registering of the associated information are indicated using the name of the element within angular brackets, as in <moderator>.

Welcome window: In the Welcome window, users provide their identification (name and password) as well as of the client and servers machines. New users give registering information in additional window.

Access window: The Access window presents options relative to Session, Users and Tools. The Session options are Create, Enter and Close. When creating a session the user specifies whether there is a moderator (<moderator>) and which tools are to active (<chat>, <whiteboard> and <vote>). When entering a session, the DocConf client generates the window corresponding to the specification for that session, and sends information to the server that the user has logged (<login> and <date>). When a user is the only one in a session, the option for closing session is activated and, if selected, this information is sent to the server, that stores the associated document.

DocConf window: The DocConf window has a child window for each toll selected to the corresponding session, as illustrated in Figure 4. The users can use the chat tool send a message to all participants (<chat>, <from> and <message>), draw or write on the whiteboard (<whiteboard>, <text>, <shape>, <polyline>), or have a voting session (<vote>). A portion of the DocConf document generated by the session illustrated in Figure 4 is shown in Figure 5.

It is important to observe, at this point, how important is the provision of a DTD for the design and

implementation of a suite such as DocConf. As suggested by [Pimentel 98], such a provision can be used to guide the development of the associated application.

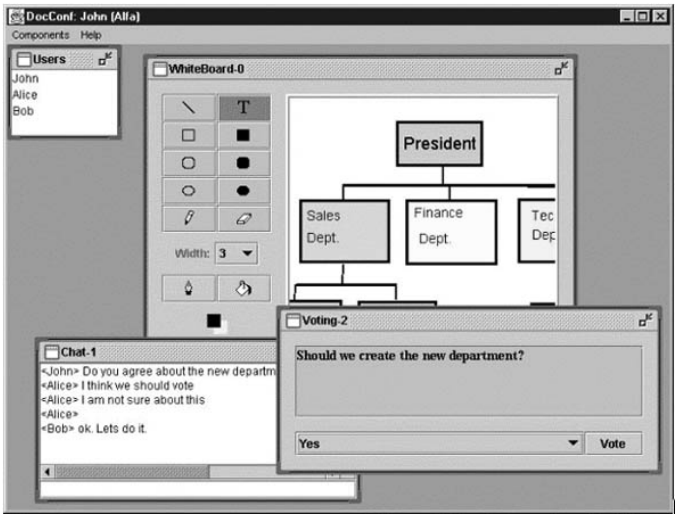

Figure 4: DocConf Client for John in session Alfa, with Alice and Bob using chat, whiteboard, and voting tools.

```
<SESSION DATE="20/02/99 18:03:25">
<HEAD>
    <ID>Alfa</ID>
    <MODERATOR>John</MODERATOR>
</HEAD>
<BODY>
<LOGIN DATE="20/02/99 18:03:36">John</LOGIN>
<LOGIN DATE="20/02/99 18:04:41">Alice</LOGIN>
<LOGIN DATE="20/02/99 18:04:45">Bob</LOGIN>
...
<WHITEBOARD COMPID="0" DATE="20/02/99 18:05:18">
    <FILLEDSHAPE TYPE="RECTANGLE" COORDS="86,121"
DIMENSIONS="93,45" THICKNESS="3"  FOREGROUND="0,0,0"
BACKGROUND="255,204,204"/>
</WHITEBOARD>
<WHITEBOARD COMPID="0" DATE="20/02/99 18:05:27">
    <TEXT COORDS="8,145" FONTSIZE="14"
FOREGROUND="0,0,0">Sales Dept.</TEXT>
</WHITEBOARD>
<WHITEBOARD COMPID="0" DATE="20/02/99 18:06:02">
    <SHAPE TYPE="LINE" COORDS="34,99"
DIMENSIONS="247,1" THICKNESS="3" FOREGROUND="0,0,0"/>
</WHITEBOARD>
...
<CHAT COMPID="1" DATE="20/02/99 18:07:08">
    <FROM>John</FROM>
    <MESSAGE>Do you agree about the new department?
    </MESSAGE>
</CHAT>
<CHAT COMPID="1" DATE="20/02/99 18:07:25">
    <FROM>Alice</FROM>
    <MESSAGE>I think we shoud vote</MESSAGE>
</CHAT>
...
<VOTING COMPID="2" DATE="20/02/99 18:08:20" ID="v2">
    <SUBJECT>Should we create the new department?
    </SUBJECT>
    <OPTION>yes</OPTION>
    <OPTION>no</OPTION>
</VOTING>
<VOTE COMPID="2" DATE="20/02/99 18:08:30"
IDREF="v2">yes</VOTE>
<VOTE COMPID="2" DATE="20/02/99 18:08:37"
IDREF="v2">no</VOTE>
<VOTE COMPID="2" DATE="20/02/99 18:08:41"
IDREF="v2">yes</VOTE>
<CHAT COMPID="3" DATE="20/02/99 18:10:08">
    <FROM>Bob</FROM>
    <MESSAGE>Acording to the votes, the new department
will be created.</MESSAGE>
</CHAT>
...
<LOGOUT DATE="20/02/99 18:12:16">Bob</LOGOUT>
<LOGOUT DATE="20/02/99 18:12:47">Alice</LOGOUT>
<LOGOUT DATE="20/02/99 18:13:19">John</LOGOUT>
</BODY>
</SESSION>
```

Figure 5: Portion of XML document generated for the session presented in Figure 4

7. Related Work

Developed by the National Center for Supercomputing Applications (NCSA), University of Illinois, Habanero provides a platform for collaboration over the Internet, specially aimed at the educational and science domains. Habanero is implemented in Java, and provides an API with a set of objects that can be used by the developer as the underlying communication mechanism to the target application. The proposal is to allow the programmer to develop a collaborative application by altering an existing single-user application or by developing a new one from scratch. Habanero provides arbitrators for applications that need locks or turn taking and, at this point, Habanero is being extended to allow asynchronous participants. DocConf shares many requirements with Habanero, in particular the aim of allowing easy extensibility and reuse of existing tools.

Prospero is a toolkit for developing groupware applications that focuses on flexibility. In this case, flexibility is understood as the range of applications that can be developed with a toolkit, which must be flexible enough to support the many ways of interactions that may occur between people [Dourish 98]. To manage the flexibility, Prospero proposes a new architectural approach - Open Implementation - where the programmer not only specifies how the application will use the toolkit infrastructure, but also can examine and modify this infrastructure implementation. In this way, the toolkit developer does not have to preview all the future uses of his system and the programmer can adapt the application to its best interest. DocConf is not implemented in such an open architecture approach; however, the component model on which it has been build provides some of the related benefits: supporting extensibility being the most important in the context of this project.

When compared to those and other work reported in the literature, DocConf is the only suite to be built upon an approach for supporting structured documents.

8. Final Remarks

In a complementary view to Engelbart's, Gaines remarks that the growth of the Web, and the evolution of their underlying technologies, may contribute to the foundations of the knowledge science [Gaines 96]: there is no doubt that facilitating human-to-human communication in such an environment may bring contributions.

We presented DocConf, an environment where users can exploit the Web as a platform to communicate using simple conferencing tools such as a shared whiteboard, a chat tool and a synchronous voting tool. A unique feature of DocConf is that the memory of a session is registered as a structured hyperdocument for later retrieval and reuse. DocConf has been built in such an approach that allows for easy extensibility and reuse of existing tools. The results provided so far with DocConf have been successful enough for having new requirements for its integration in a project related to computer supported collaborative learning. Other scheduled tasks are the full provision for asynchronous sessions and support to other media.

Acknowledgement: Marcos Kutova was sponsored by FAPESP-SP- Brazil.

References

[Bosak 98] Bosak, J. XML, Java, and the future of the Web. Sun Microsystems (c. 1 Oct 98) http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm (1997).

[Benford 97] Benford, S., Greenhalgh, C. & Lloyd, D. (1997). Crowded Collaborative Virtual Environments. *CHI'97 – Conference on Human Factors in Computing Systems, 1997*. ACM. 59-66.

[Chabert 98] Chabert, A., Grossman, E., Jackson, L. S., Pietrowiz S. R. & Seguin, C. (1998). Java object-sharing in Habanero. *Communications of the ACM*, 41 (6), 69-76.

[Connolly 98] Dan Connolly. The XML Revolution http://helix.nature.com/webmatters/xml.html

[CSS 96] W3C (1996) Cascading Style Sheets, level 1 (CSS1). http://www.w3.org/TR/REC-CSS1/

[Dourish 98] Dourish, P. (1998). Using Metalevel Techniques in A Flexible Toolkit For CSCW Applications. *ACM Trans. Computer-Human Interaction*, 5 (2), 109-155.

[Engelbart 90] Engelbart, D.C. (1990). Knowledge Domain Interoperability and an Open Hyperdocument System. *CSCW'90 – Conference on Computer Supported Cooperative Work, 1990*. ACM. 143-156.

[Engelbart 95] Engelbart, D.C. (1995). Toward Augmenting the Human Intellect and Bosting our Collective IQ. *Communications of the ACM*, 38 (8), 30-33.

[Engelbart 98] Engelbart, D.C. (1998). Augment, Bootstrap Communities, the Web: What Next? *CHI'98 – Conference on Human Factors in Computing Systems, 1998*. ACM. 15-16.

[Glushko 98] Glushko, R.J., (1998) The XML Revolution. Proc. *WebNet'98*. AACE. 342-344.

[Gaines 96] Gaines, B.R., (1996). Convergence to the Information Superhighway. *WebNet'96*. AACE.

[Pimentel 98] Pimentel, M.G., Santos, J. & Fortes, R. (1998). Tools for Authoring and Presenting Structured Teaching Material in the WWW. *WebNet'98*. AACE. 194-199.

[Rada 98] Rada, R., Cargill C. & Klensin, J. (1998). Consensus and the Web. *CACM*. 41 (7), 17-22.

[Streitz 91] Streitz, N. (1991) The Role of Hypertext for CSCW Applications. *Hypertext'91*. ACM. 369-375.

[XML 97] W3C (1997). The Extensible Markup: XML 1.0. http://www.w3.org/XML/

[XSL 99] W3C (1999). The Extensible Markup: Extensible Stylesheet Language (XSL) Specification. http://www.w3.org/TR/WD-xsl/