

JADE: Jabber-based Authoring in Distributed Environments

Andrew Rocznik
Multimedia Communications Research
Laboratory (MCRLab)
University of Ottawa
Ottawa, Canada
roczniak@mcrmlab.uottawa.ca

Abdulmotaleb El Saddik
Multimedia Communications Research
Laboratory (MCRLab)
University of Ottawa
Ottawa, Canada
abed@mcrmlab.uottawa.ca

ABSTRACT

We present our initial results in developing a framework for collaborative multimedia authoring tools. This research is motivated by the lack of tools that take into account consumers' quality of *experience*. By mapping factors that have an impact on the quality of experience into requirements, we are developing a framework for tools that allow retrieval and manipulation of multimedia objects, and collaborative authoring of multimedia documents based on Jabber set of protocols.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Evaluation, methodology; H.5.2 [User Interfaces]: Interaction styles; H.5.3 [Group and Organization Interfaces]: Computer-supported cooperative work

General Terms

Design, Human Factors, Performance

Keywords

Multimedia authoring, Quality of experience, Jabber

1. INTRODUCTION

We are building JADE, a framework for collaborative multimedia authoring that will provide tangible Quality of Experience (QoE) to users. The ubiquity of computers, networks and multimedia objects means that we are increasingly interacting with multimedia documents. For these multimedia documents to be authored in a collaborative manner, new tools that are user-friendly are necessary. Users of these tools are envisioned to be creative content developers, and everyday users of the internet. Users will try and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'05, November 6–11, 2005, Singapore.
Copyright 2005 ACM 1-59593-044-2/05/0011 ...\$5.00.

Factors	Requirements
Join a community whose interests are multimedia authoring	Network-centric application ¹
Store, share and access multimedia objects	Remote database access
Authored document belongs to all collaborators	Decentralized architecture and replication
Joining and leaving groups is easy	Ad-hoc collaboration, support for late-join
Many collaborators can form a group	Scalable infrastructure, minimal deployment effort
Users are unaffected by others joining or leaving	Performance and robustness ²

Table 1: Factors that influence QoE and their corresponding requirements

adopt such tools, only if their experience with them is positive. Although each user's level of familiarity and comfort with computing environment is different, there are nonetheless common factors that can allow us to quantify their experience in a single measure of QoE. This measure is a composite of factors presented in Table 1. We can then map those factors to functional requirements. Those requirements will in turn drive decisions on the selection of technologies used in building JADE, will motivate the functionality offered by our authoring framework, and finally, will help in defining experiments necessary for validation purposes.

We created Table 1 by loosely following the method outlined in [8] to identify the tasks (factors), and to create a task/needs matrix; note however that our factors do not relate to the specific nature of various multimedia objects. Factors that influence the QoE were gleaned from discussions with various research members of our lab (their perception of a *good* authoring tool), and we are working on a more rigorous questionnaire approach for future iterations.

¹We will start with a networked environment and build a tool, rather than making an authoring tool networked.

²There should be no noticeable effects when the size of the collaborating group increases. Similarly, the state of the document is unaffected by leaves, joins and clients' failures.

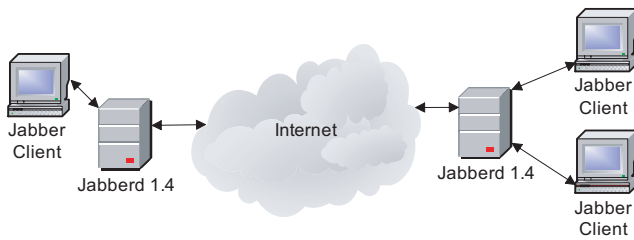


Figure 1: Jabber network consists of Jabber servers (running Jabberd 1.4 in our case) and Jabber clients that register with a specific server.

Jabber [1] is a set of streaming XML protocols that enable participants to exchange structured information in close to real time. Two of those protocols in particular are the extensible messaging and presence protocol (XMPP) defining messaging, presence, and request-response services between any two network endpoints, and instant messaging and presence protocol (XMPP-IM) defining instant messaging and presence functionality. Jabber does not require any specific network architecture, but following examples of many instant messaging applications, we are implementing a simple client/server architecture (see Fig. 1). Each client registers with a server, which acts as a client proxy on the network. Communication on the network uses DNS for address resolution. We are using Jabberd 1.4 software for Jabber servers, and we build our own Jabber client based on the Smack API.

Although traditionally used for instant messaging application, Jabber provides a general framework that can be leveraged in multimedia authoring applications. Building on top of Jabber, functionalities such as notification, negotiation, data exchange, addressing, presence, information metadata and support for multi-party exchange do not have to be recreated for our application. Furthermore, and perhaps the most important motivating factor, Jabber allows *conversations* between people as well as between applications, or between a person and an application. What that means is that the same framework can be used to support collaboration (person to person conversation), while concurrently and transparently allow synchronization of authoring applications (application to application conversation). Being open-source, Jabber has also the advantage of not being restricted to a particular platform or product, making integration of heterogeneous machines easy.

Alternative technologies include the JXTA framework and WebDAV set of protocols. We have decided to build on Jabber since JXTA, although very flexible does not meet our requirement of minimal effort necessary to deploy authoring applications, and WebDAV which does not meet our decentralized architecture requirement.

The rest of the paper is structured as follows. In section 2 we present the JADE framework functionality while experiments are presented in section 3. Section 4 concludes.

1.1 Related Work and Contribution

Collaborative environments are well understood by the scientific community [9]. Similarly, many authoring tools are based on the standardized synchronized multimedia integration language (SMIL). The combination of those two

areas was explored in [12]. The authors present a system that allows users to collaborate on a multimedia presentation through a unified spatial and temporal information interface. In contrast to our framework, it requires an application specific TCP-based protocol, and the emphasis is placed on the authoring tool that is subsequently interfaced with communication and concurrency control modules. Jabber based applications are plentiful and include tools for chatting, blogging and email notification, but up until now, none that target collaborative authoring.

To the knowledge of the authors, no previous attempt has been made to develop a distributed multimedia authoring framework based on Jabber. Although the underlying technologies are known, this specific combination of tools and approaches is novel.

2. IMPLEMENTATION DETAILS

JADE is currently composed of two components, Jabber Data-base Access (JDA) and an interactive J3D component. Both components are interfaced to Jabber and to the application-specific code.

As Jabber is designed for near real-time communication, the size of data transferred and frequency of transfers is limited in order to allow fair sharing of Jabber router resources. This makes an additional out-of-band communication necessary to transfer large amount of data between clients, and should be implemented with well-known protocols. The schematic is presented in Fig. 2. As can be seen, two clients communicate through an intermediary Jabber router which directs Jabber stanzas³ to their appropriate recipients. JDA allows a client to access a database, which can be located locally, or at another client. JDA allows users to negotiate access policies to each other's databases; and does not place restrictions on what is stored in databases; larger objects should be automatically transferred using out-of-band protocols. These transfers can use semantics defined in JEP-0096 (File Transfer) for example. The J3D component is built on top of Java 3D API and implements authoring and collaboration metaphors (as explained below) that allow users to interactively author a multimedia document. Metaphors are application agnostic, while semantics of object placement (e.g. spatial or temporal) need to be defined and understood by all participants prior to collaboration - and are implemented at the application-specific layer.

The application-specific code contains the implementation for multimedia object editing, manipulation, and building relationships between them. This code is also responsible for enforcing all the rules specific to the document being created. In our proof-of-concept authoring tool, six people work on the same puzzle consisting of images. This application will verify if a particular placement of an image will not conflict with another object, possibly placed by a different user.

2.1 Authoring

2.1.1 Private Window Metaphor

The private window is a personal space where a user adds, removes and orders multimedia objects. Ordering can be spatial, temporal or both, as defined by application-specific

³A stanza is a discrete semantic unit of structured information sent between any two entities over a network.

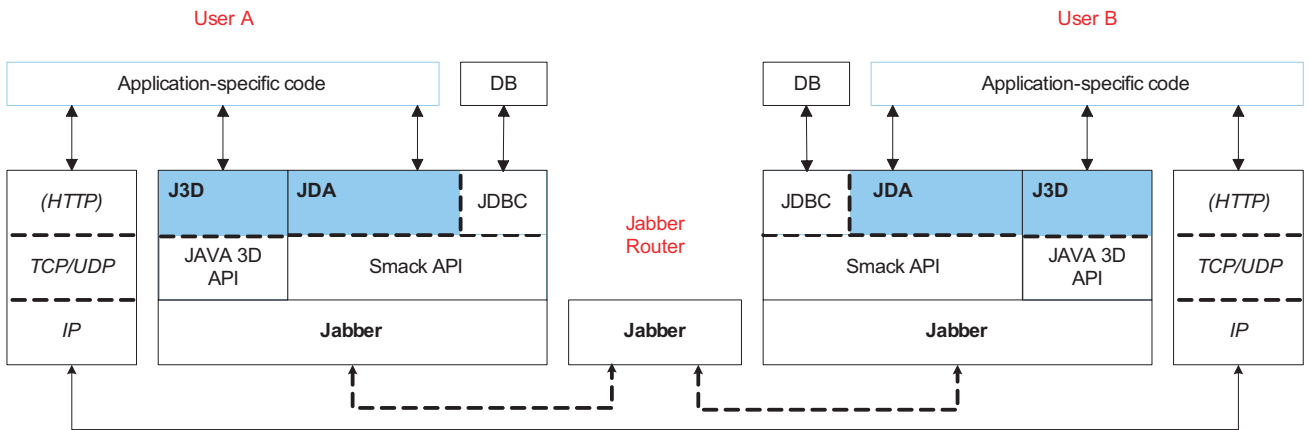


Figure 2: JADE framework components, highlighted in blue; an emphasis is placed on separation of application-specific code and JADE components

code. Objects can be private, or obtained from another participant using the JDA component. This, in fact, is a canvas for users to author multimedia documents. In our prototype application, objects are ordered spatially by the user in such a way as to match (in a puzzle-like way) what other participants are doing.

2.1.2 Public Window Metaphor

The public window is a public collaboration space. This is where the interaction takes place, where users become aware of other participants' contributions. This space is application-specific and there should be no restrictions on what this space represents. Current implementation uses a three-dimensional cube where objects are arranged in spatial relationship to each other and where each face belongs exclusively to one participant.

2.2 Collaboration

Any transfer or update from the private window to the public window implies that data representing objects will be shared with other participants. Depending on the application and the object itself, a different method should be employed to conduct the transfer. In the case when objects are well-known and available to all participants, an identification of the object needs to be transmitted, which can be done over the Jabber network. If the objects are large, such as video clips for instance, and some participants do not have access to them, those objects then need to be transmitted out-of-band using protocols such as HTTP or FTP. Current prototype implementation uses the former method, where the identification and the position of the object are sent encapsulated in a Jabber stanza.

Depending on the number of participants, we have two distinct communication models. In the case of two users, a *peer-to-peer* metaphor is implemented, whereas for three or more participants, a *conference room* metaphor is implemented by the J3D component. In the former case, users perceive the communication as a bi-directional *conversation*. On the network level, this is equivalent to a *peer-to-peer* architecture where both users simultaneously act as client and server. In the latter case, a user that transfers contents of its private window to the public window perceives the com-

munication as a broadcast to a group of interested users. On the network level, this is similar to a multicast.

2.2.1 P2P Metaphor

Users create and access shared data, potentially resulting in global inconsistency if proper concurrency control mechanisms are not implemented. JADE needs to address two incompatible requirements [11]:

- High responsiveness. Perception of quality depends on the timeliness of response of the system to other users' actions.
- High concurrency. Concurrent actions of users do not result in a globally inconsistent state, even in the presence of conflicts.

Propagation mechanisms (the actual transfer) are separated from propagation policies. A propagation policy determines when updates should be propagated [10], and depends on specific requirements of the application. In our prototype, the user decides when to update the public window. If floor-control techniques [5] are not implemented, conflicts will arise, and a mechanism to resolve them has to be provided. Conflict resolution can be automatic, manual or can be based on data or application semantics. In our prototype, we currently restrict users to their own exclusive space in the public window.

2.2.2 Conference Room Metaphor

Here, just as in the P2P metaphor, consistency of data needs to be ensured. Furthermore, when large objects are transferred to multiple recipients, an efficient multicast is needed. Perception of quality will depend on reliability and timeliness of transfers. Our prototype currently has basic transfer abilities, and most of our effort in future iterations will concentrate on:

- Evaluating two models, the process group model and the transaction model. The process group model treats the distributed system as a collection of users that communicate by sending multicast messages. Techniques for ordering concurrent messages as well as ensuring atomicity of message delivery in the presence of

delays and link failures, and changes to group membership need to be addressed [3]. The transaction model is characterized by operations being grouped into transactional units, with techniques available for ordering concurrent transactions and ensuring their atomicity [7].

- Implementing an efficient multicast between collaborators. Interactive applications, where interactivity is defined as the ability to change the state of an application by supplying external events (such as user actions in a whiteboard application or networked games), can use RTP/I [6]. Recent research in Application-Layer Multicast also presents an interesting opportunity [4].
- Security issues, including authentication, authorization, data confidentiality, data integrity and possibly non-repudiation. Additionally, flexibility and scalability characteristics of any solution need to be addressed [2].

3. EXPERIMENTS

We have implemented a proof-of-concept authoring tool for six people working on the same puzzle. This will form our basis for creating experiments that validate our system requirements. We present here a brief description of experiments that will ultimately give us an idea of the quality of experience as perceived by the user:

- Finding a community interested in multimedia authoring. We will look into the suitability of JEP-0030 (Service Discovery) and JEP-0045 (Multi-User Chat) protocols to support finding, negotiating and joining an authoring community, and into creating specialized server-side implementations of Jabber components.
- Accessing multimedia objects. Assessing the performance (delay and integrity of data) when downloading large media files is paramount from users' perspective. The tool could be deemed unusable if sharing of objects takes an inordinate amount of time, or if the data received is corrupted.
- Stability and reliability of the framework in the face of join/leaves, and integrity of the document and authoring session. This experiment will attempt to provide bounds for the rate of join and leaves within which the tool preserves the integrity of data.
- Scalability. This experiment will attempt to provide bounds for the number of concurrent participants in an authoring session. These bounds are expected to be driven in large part by the two preceding experiments.

4. CONCLUSION

Our current effort started as an attempt to create an authoring infrastructure requiring a minimal effort to deploy, maintain and scale-up as the number of users increase. We have decided to build a framework based on Jabber - JADE, allowing for more complex authoring applications to be built on top of it. Keeping the end-users in mind, we have decided that our functional requirements should be dictated

by the experience as perceived by the user when using applications built on JADE. A basic proof-of-concept application has been developed, allowing up to six users to collaboratively put together a three-dimensional puzzle. Many issues remain, notably consistency of data, efficient multimedia object dissemination and security, as well as performing appropriate experiments in order to validate requirements, which, ultimately, will give us an idea of how well JADE meets user's needs.

5. ACKNOWLEDGMENTS

The authors acknowledge the design and implementation contributions of Vanessa Ferreira, Jacek Pakulski and Huaizhou Ding, as well as the financial support of Natural Sciences and Engineering Research Council of Canada and the Government of Ontario (PREA).

6. REFERENCES

- [1] *www.jabber.org*.
- [2] D. Agarwal, K. Jackson, and M. Thomson. Securing collaborative environments. In *Proc. Workshop on Advanced Collaborative Environments*, pages 125–133, July 2002.
- [3] J. Chang and M. K. Makemchuk. Reliable broadcast protocols. *ACM Transactions on Computer Systems*, 2:251–273, August 1984.
- [4] Y.-H. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Measurement and Modeling of Computer Systems*, pages 1–12, 2000.
- [5] H.-P. Dommel and J. J. Garcia-Luna-Aceves. Efficacy of floor control protocols in distributed multimedia collaboration. *Cluster Computing*, 2(1):17–33, 1999.
- [6] M. Mauve, V. Hilt, C. Kuhmunch, and W. Effelsberg. Rtp/i - towards a common application level protocol for distributed interactive media. *IEEE Transactions on Multimedia*, 3(1), 2001.
- [7] J. Munson and P. Dewan. A concurrency control framework for collaborative systems. In *Proc. ACM CSCW*, pages 278–287, 1996.
- [8] A. Patrick, J. Singer, B. Corrie, S. Noël, K. E. Khatib, B. Emond, T. Zimmerman, and S. Marsh. A qoe sensitive architecture for advanced collaborative environments. In *QSHINE*, 2004.
- [9] E. M. Schooler. Conferencing and collaborative computing. *Multimedia Systems*, 4(5):210–225, 1996.
- [10] H. Shen and C. Sun. Flexible notification for collaborative systems. *Proc. ACM Conference on Computer Supported Cooperative Work*, November 2002.
- [11] C. Sun and D. Chen. Consistency maintenance in real-time collaborative graphics editing systems. *ACM Transactions on Computer-Human Interaction*, 9(1):1–41, March 2002.
- [12] M. Sung and D. Lee. A collaborative multimedia authoring system. *Lecture Notes in Computer Science*, Springer-Verlag, 3033:311–318, 2004.