

Instant Collaboration

Using Context-Aware Instant Messaging for Session Management in Distributed Collaboration Tools

Klaus Marius Hansen and Christian Heide Damm
 Centre for Pervasive Computing
 Department of Computer Science, University of Aarhus
 Aabogade 34, 8200 Aarhus N, Denmark
 {marius, damm}@daimi.au.dk

ABSTRACT

Distributed collaboration has become increasingly important, and instant messaging has become widely used for distributed communication. We present findings from an investigation of instant messaging use for work-related activities in a commercial setting. Based on these findings, we propose a lightweight session management design for distributed collaboration tools based on context-aware instant messaging. An implementation of this design is presented and an ongoing evaluation is discussed.

Keywords

Instant messaging, computer-supported co-operative work, distributed collaboration tools, interaction design.

INTRODUCTION

Distributed collaboration is becoming more prevalent and important (CNN 2001). A vital part of most distributed collaboration tools is *session management* (Beaudouin-Lafon 1999). Sessions are ephemeral couplings of users involved in common activities, and session management involves, among others, creating sessions, joining sessions, leaving sessions, and providing awareness of ongoing sessions.

This paper presents a user interface for session management that builds on existing tools, viz., *instant messaging* (IM), in order to provide a lightweight way of managing sessions. IM clients enable users to chat with other users by providing means for IM of short text messages.

Figure 1 shows an example of an IM client (Yahoo!; <http://messenger.yahoo.com>). The window to the right shows the statuses of klaus_hansen2001's friends and the window to the left shows an ongoing IM session between klaus_hansen2001 and avratzer. In the Yahoo! messenger, you double-click on an active friend in order to start a chat session with that friend.

We are considering session management in the context of the Knight Tool:

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.

NordiCHI 10/02 Århus, Denmark

© 2002 ACM ISBN 1-1-58113-616-1/02/0010...\$5.00

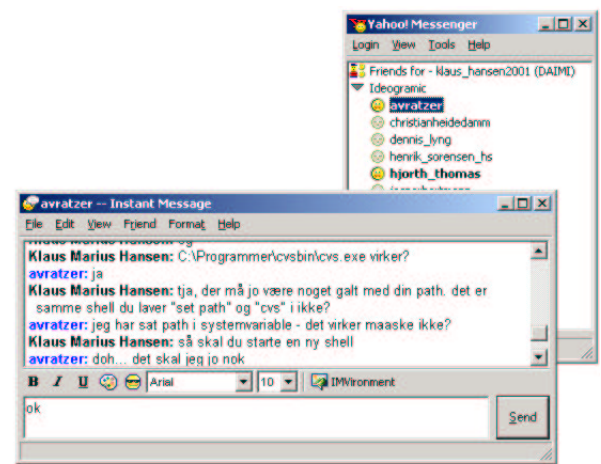


Figure 1: An Instant Messaging Client and Session

The Knight Tool

The Knight tool (Damm, Hansen & Thomsen 2000) supports co-located, collaborative object-oriented modelling. Object-oriented modelling is concerned with creating program representations of phenomena and concepts in a problem domain as *objects* and *classes* and involves a diverse set of competencies (Christensen, Crabtree, Damm, Hansen, Madsen, Marqvardsen, Mogensen, Sandvad, Sloth & Thomsen 1998).

The main use of Knight is on electronic whiteboards — large pressure-sensitive computer screens — which enable users to co-operate and coordinate when co-located. Figure 2 shows this to left. To the right, the figure shows the main way of interacting with Knight through the use of pen gestures: a pen stroke is recognized as a box gesture (left) and consequently, a Unified Modeling Language (UML; (OMG 2001)) class symbol representing a modelled concept is created in Knight (right). The most common operations of creating, moving, resizing, and deleting UML elements can be performed using gestures. This provides a fast and direct way of interacting with Knight (Damm et al. 2000). Gestural interaction is also particularly well-suited to electronic whiteboards (Pedersen, McCall, Moran & Halasz 1993).

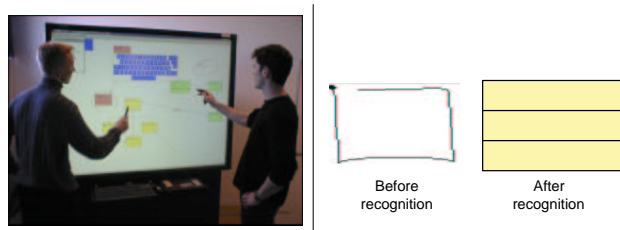


Figure 2: Left: Use of Knight on an Electronic Whiteboard. Right: Recognition of Pen Strokes in Knight

The Knight Tool has been commercialised as *Ideogramic UML* by Ideogramic ApS (<http://www.ideogramic.com/>).

USER STUDIES

Based on anecdotal observations of our own use of computer-mediated communication, we decided to investigate further the use of IM in software development activities. We had been geographically separated from a software development group, with which we collaborated closely, for a long period of time. During this period, we had used IM extensively for communication. At the same time, we had begun building a version of the Knight Tool that supports distributed collaboration. Thus, we focussed on the use of IM to support distributed collaboration activities.

We studied a development group at Ideogramic ApS and their use of IM. The group, of which the authors are part, primarily work at three different locations: Ideogramic's offices, University of Aarhus, and from the homes of the developers. Three out of seven developers changed location frequently, while the last four developers mainly worked out of Ideogramic's offices.

IM use was studied in three ways through:

- *Observations.* We made informal observations of developers at Ideogramic while they were using IM.
- *Transcripts.* We obtained logs of IM sessions from developers at Ideogramic and analysed these textually.¹
- *Interviews.* We conducted open-ended, qualitative interviews with developers, focussing on level and patterns of use of IM in collaborative software development activities.

Some of the main characteristics of the analysed material are summarised in the following sections:

Different Levels of Coupling

IM can be nearly synchronous: senders write short messages that are instantly transmitted to the receiver if the receiver is active. The following quote from an interview shows how the written communication form also allow receivers to ignore a message or postpone the answer of it:

¹In the following, the names of participants are changed for the sake of anonymity, and the text has been translated from Danish into English. The areas of particular interest in the transcripts have been emphasized.

IM works really well when people are not in the same place ... people are able to answer when they have the time ... using the telephone they need to attend to it right away. Sometimes I start using IM and later use the telephone if necessary

The following transcript of a software design situation exemplifies this and furthermore shows how IM allows for a shift to other more synchronised communication forms

(11:13) L: What's the idea with the [...] list for each [...] object? [...]

(11:14) D: Hmmh. I'm not sure?

(11:14) D: If it is redundant information, it shouldn't be there [...]

(...)

(11:19) D: [...] I think K knows more about it than I do [...]

(11:22) L: It looks like it is being used [...]

(11:22) D: Try to call K at (+1) 404/872-0849

(11:22) D: yes, it is used [...]

(...)

(11:23) L: when does a [...] object have multiple [...] lists?

(11:24) D: yahoo?

Following this, the Yahoo! IM client was used to create a voice connection over the Internet. In this way, the IM client provided for a switch from written IM communication to verbal IM communication. This shows a characteristic way in which a loosely coupled IM session is turned into a coupled voice chat session. The difference shows in the beginning of the transcript in which there is a pause of four minutes (11:19–11:22) between two messages.

Often developers went further and tried manually to synchronise text editors or the use of the application they were developing. In many ways, IM represents a hybrid between written and verbal communication and also allows for communication patterns that are biased towards one or the other (Volda, Newstetter & Mynatt 2002).

Co-Construction and Coordination

IM was extensively used to collaboratively identify and solve problems and to create new designs:

I use IM for solving problems ... I contact the person who knows what I need to know ... particularly if I need an answer fast

Also, people were often used by people at other locations to help coordinate or to synchronize work:

(19:49) K: i have a status question: did you manage to make a new software release this week?

(19:49) T: No :o(There were some problems [...]

(19:50) K: ok. so i will just fix bugs [...] (the other things, i should do [...] require larger software restructurings...)

(19:51) T: Great! How is it going anyway?

Often e-mails were used to transfer, e.g., design documents. These documents were then used at both sides for reference during the following IM discussions. Simple design decisions were also frequently made based on a few instant messages.

Context Awareness

Awareness of where people were and which activities they were engaged in were seen as important:

status indicators [showing whether people are, e.g., available or busy] are really useful — if people keep them up-to-date ... the places where people are, are also significant ... are they at home or at the university? It is important to be able to see if people are working

Most IM clients provide a way of seeing whether people have used their computers recently. Showing which activities people are engaged in or whether they are available or busy, on the other hand, has to be done manually by changing status messages in the IM clients. This often leads to misrepresented statuses:

(08:16) M: I just recalled that we have to pay H for his extra work hours. Did you remember to do that?

(08:19) M: ??

(08:20) M: (ohh, you're having the meeting. I'm off.)

DESIGN

Distributed Knight is an extension of the Knight tool with support for synchronous, distributed collaboration facilities. Using Distributed Knight, users at separate physical locations are able to work synchronously on the same modelling problem. The implementation is based on a peer-to-peer architecture in which multiple Distributed Knight clients may replicate data and collaborate on this data in real time (Damm & Hansen 2002).

Based on our user studies outlined above, we decided to build an IM client that would integrate with Distributed Knight and support lightweight session management analogous to the way in which standalone Knight supports lightweight co-located collaboration.

Session management should support users in creating, maintaining, and deleting collaboration relationships in distributed collaboration. An IM client seems well-suited for this in that it:

- supports an entry *level of coupling* appropriate for initial discussions. Furthermore, it provides ways to increase or decrease levels of coupling between collaborators as appropriate
- provides means for negotiating *coordination and co-construction*
- may provide *context awareness* of locations and activities of people but also of ongoing sessions

Figure 3 shows the current implementation, *AwareMessenger*, of an IM client providing the session management interface for Distributed Knight. The Aware Mes-

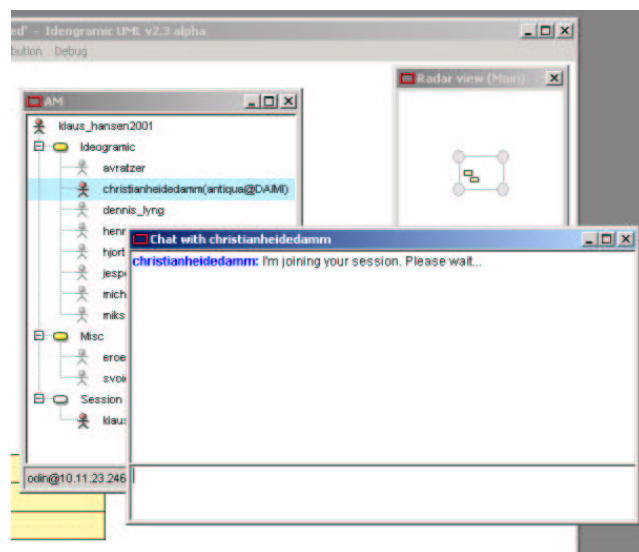


Figure 3: AwareMessenger

senger client shows activity level and location of users. In the figure, only klaus_hansen2001 and christianheidedamm are active. Klaus_hansen2001 has started a session using Aware Messenger based on a problem that has surfaced through the discussion with christianheidedamm and has now, through Aware Messenger invited christianheidedamm to join this session. The bottom of the Aware Messenger client shows a *transient session group* containing the members of the session klaus_hansen has started, currently only klaus_hansen2001. Users may now operate on this session group, e.g., join the session.

When christianheidedamm joins the session, discussions can continue in Aware Messenger while synchronous collaboration is taking place in Distributed Knight. Next steps in coupling that we are currently implementing and that fit well in an IM client is support for voice and video conferencing.

Awareness of places and things are implemented through an integration with the Tag'N'Track (TNT) framework (Ørbæk 2002) supporting location-dependent services and tracking of work material. Currently, we provide information on the location of the computer each IM client runs on. For the future, we plan to take further advantage of this by providing extensive awareness of people, places, and things.

PRELIMINARY EVALUATIONS

Aware Messenger has been implemented and integrated into Distributed Knight which is currently functional but still under development. The present version is used to conduct pilot studies of various aspects of Distributed Knight including its support for session awareness and management. We have set up evaluation sites at the University of Aarhus and at Ideogramic where members of various development groups are actively using Distributed Knight, and are starting to provide both sites with voice and video communication equipment. So far the results of these evaluations are positive.

These pilot evaluations aim at making Distributed Knight usable in real, production use. The second part of the user study will then aim to observe distributed collaboration concurrently at both sites and combine this with quantitative information through use logs which are already established.

FUTURE AND RELATED WORK

As mentioned above, our current work focuses on making Distributed Knight so complete that it can be used in everyday production work by a small group of developers. With respect to interaction, this involves primarily adding additional awareness information to Knight such as awareness of what other session users are doing in a session. Based on Distributed Knight, we plan to generalize our code platform to create a framework for building distributed collaboration tools handling highly structured data similar to UML data. This generalization will also involve a reuse and generalization of the interaction with Distributed Knight and thus also of the Aware Messenger client.

Recently, a large body of HCI research has investigated IM and related kinds of computer-mediated communication (e.g., Herbsleb, Atkins, Boyer, Handel & Finholt 2001). Most of these focus on recreational use of IM, but workplace studies are emerging (Bradner, Kellogg & Erickson 1999, Nardi, Whittaker & Bradner 2000, Herbsleb et al. 2001). Also, commercial workplace uses of IM is becoming available: the Groove desktop collaboration platform (<http://www.groove.net>) integrates an IM client. To our knowledge, however, the session management use of IM in Distributed Knight is the first attempt to closely integrate an IM application with a (distributed collaboration) tool in order to build upon the inherent characteristics of IM.

SUMMARY

This paper has presented the session management component of the distributed collaboration tool *Distributed Knight* for synchronous, distributed collaboration in object-oriented modelling. The design of the session management component has been based on observations of and interviews with software developers and has resulted in an instant messaging client supporting users in creating, joining, and leaving sessions through an emphasis on support for levels of coupling and context awareness. The instant messaging client has been implemented and is currently being evaluated in a series of user studies.

REFERENCES

Beaudouin-Lafon, M., ed. (1999), *Computer Supported Cooperative Work*, Wiley.

Bradner, E., Kellogg, W. & Erickson, T. (1999), The adoption and use of Babble: A field study of chat in the workplace, in 'Proceedings of ECSCW'99, Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work'.

Christensen, M., Crabtree, A., Damm, C., Hansen, K., Madsen, O., Marquardsen, P., Mogensen, P., Sandvad, E., Sloth, L. & Thomsen, M. (1998), The M.A.D. experience: Multiperspective Application

Development in evolutionary prototyping, in 'Proceedings of the ECOOP' 1998', pp. 13–40.

CNN (2001), Telework rend rooted in convenience. <http://www.cnn.com/2001/TECH/ptech/11/12/-telework.comdex.idg/index.html>.

Damm, C. & Hansen, K. (2002), Distributing Knight – Using type-based publish/subscribe for building distributed collaboration tools. In Proceedings of NWPER 2002.

Damm, C., Hansen, K. & Thomsen, M. (2000), Tool support for object-oriented cooperative design: Gesture-based modeling on an electronic whiteboard, in 'Proceedings of CHI 2000, ACM Conference on Human Factors in Computing Systems', pp. 518–525.

Herbsleb, J., Atkins, D., Boyer, D., Handel, M. & Finholt, T. (2001), Introducing instant messaging and chat in the workplace, in 'Proceedings of ECSCW'01, Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work'.

Nardi, B., Whittaker, S. & Bradner, E. (2000), Interaction and outeraction: Instant messaging in action, in 'Proceedings of CSCW'2000, Proceedings of the Conference on Computer-Supported Cooperative Work', pp. 79–88.

OMG (2001), Unified Modeling Language specification 1.4, Technical Report formal/01-09-67, Object Management Group.

Pedersen, E. R., McCall, K., Moran, T. P. & Halasz, F. G. (1993), Tivoli: an electronic whiteboard for informal workgroup meetings, in 'Proceedings of CHI 1993, ACM Conference on Human Factors in Computing Systems', pp. 391–398.

Ørbæk, P. (2002), Tag'n'Track framework. <http://www.daimi.au.dk/workspace/>.

Voida, A., Newstetter, W. & Mynatt, E. (2002), When conventions collide: The tensions of instant messaging attributed, in 'Proceedings of CHI 2002, ACM Conference on Human Factors in Computing Systems', pp. 187–194.