The integration of synchronous communication across dual interaction spaces

Martin Mühlpfordt

Fraunhofer IPSI
Dolivostr. 15
64289 Darmstadt
Martin.Muehlpfordt@ipsi.fraunhofer.de
Translated by Gerry Stahl

Abstract: Dual interaction spaces-that combine text chat with a shared graphical work area-have been developed in recent years as CSCL applications to support the synchronous construction and discussion of shared artifacts by distributed small groups of students. However, the simple juxtaposition of the two spaces raises numerous issues for users: How can objects in the shared workspace be referenced from within the chat? How can users track and comprehend all the various simultaneous activities? How can participants coordinate their multifaceted actions? We present three steps toward integration of activities across separate interaction spaces: (1) support for deictic references, (2) implementation of a history feature and (3) display of social awareness information.

1 Introduction

The construction, modification, annotation and arrangement of shared artifacts are key activities in many collaborative learning settings. Software systems now exist that permit synchronous coordinated manipulation of such shared artifacts even for geographically distributed users, by providing a shared graphical workspace.

A shared workspace in a collaborative environment is an area of the software interface that allows a participant to construct and manipulate a graphical object so that the object and the effects of the manipulation appear in the corresponding area of the other participants' interfaces, essentially in real time [**?**]. These shared workspaces may be used for creating and using external representations of knowledge [**?**], for collaboratively completing design tasks [**?**] or for working together with simulations [**?**, **?**]. The design of shared workspaces is an important topic in computer-supported collaborative learning (CSCL).

Learning at a distance requires a medium of communication. The medium can be auditory, audio-visual[1] or text-based. For collaborative learning, textual synchronous communication-chat-has two main advantages: For the chat poster, writing encourages a more careful plan-

[1] It is perhaps telling that there is in general no advantage to adding a video channel to purely auditory communication [**?**].

ning of one's contribution; it fosters reflection on the discourse. For the recipient, the communication is persistent and available in symbolic form that "may be searched, browsed, replayed, annotated, visualized, restructured and recontextualized" [**?**].

The combination of a shared workspace with chat makes two regions for interaction available to a group in the form of a dual interaction space [**?**]. The chat provides a medium of communication for the exchange of textual messages; the shared workspace allows for the collaborative construction and manipulation of shared artifacts that are relevant to the task at hand. In most groupware systems for synchronous distance learning, the chat and graphical workspace simply appear next to each other as two visually distinct areas of the application that are largely functionally independent of each other. This introduces a number of problems for the users [**?**, **?**]. For instance, if a group of students want to create a concept map in the shared workspace consisting of arguments pro and con and their relationships to each other, this raises the following questions:

1. How can objects and relationships within the workspace be referenced from a posting in the chat area?
2. How can the participants grasp and understand the relationships among each other of the activities and messages that are part of a single collaborative interaction but are distributed across the two interaction spaces? E.g., how can one establish that the message, "I agree," is a response to the introduction of a particular new node in the argumentation graph?
3. How can the participants coordinate their actions in the graphical workspace and in the chat with each other? E.g., when and by whom should an argument introduced in the chat be added to the concept map?

A better software integration of chat and workspace is needed to overcome such difficulties [**?**, **?**, **?**]. This paper will propose integration measures for three relevant aspects of the connection of chat and shared workspace. Section 2 analyzes these problems. Section 3 describes the integration measures. Then in Section 4 a collaboration environment named ConcertChat will be presented which implements these measures. Section5 discusses related work and systems. Finally, Section6 reports on results with ConcertChat and suggests questions for future work.

2 Problems in combined interaction spaces

A shared workspace can play at least two contrasting roles within a collaborative session. It can, for instance, provide the central location for the joint activity of the participants, with the chat playing a supportive role in discussing and disambiguating the activities that take place in the workspace. Conversely, the chat discourse can dominate, with the graphical workspace serving as a resource for clarification or for illustrating things that are hard to articulate in words. Which way communication is divided between the dual spaces depends upon the current task, the meta-communicative skills of the participants and the respective affordances of the two media [**?**, **?**]. In any case, the activities in the chat and the shared workspace are typically intimately interrelated.

A prominent characteristic of chat is the delay between the production of a message by its author and its presentation to others when it is complete. This has two main advantages: that the author can revise the message before sending it [**?**] and that several people can be producing messages at the same time, unlike in spoken conversation. However, it also leads to the constant danger of incoherence, which forces the participants to work additionally on explicitly coordinating the content and structure of their interactions. The problem is that, unlike in conversation, in chat, responses often do not immediately temporally follow the messages to which they are responding. The coherence of interaction is highly dependent upon the response structure between messages. But in the time it takes for someone to prepare and send a response to one note, a note from someone else can be posted, causing "interrupted turn adjacency" [**?**]. A number of specific communication strategies may be evoked to deal with this [**?**, **?**]. In order to minimize the delay in responding, mistakes in syntax and wording are accepted and many abbreviations or acronyms are used. To make references explicit, the addressee to whom one is responding may be explicitly named.

The fact that several people can be producing messages at the same time means that the common conversational rules of turn-taking do not apply. The resulting parallelism can scarcely be avoided, and must particularly be taken into account when multiple topics are discussed simultaneously. This problem is eased by the fact that the flow of chat is documented in the persistent transcript,[2] which is visible-at least for the last several postings. The chat window serves not only as the location of communications, but also as a representation of the temporal order of the messages. In contrast, the graphical workspace usually only shows the current state. All information about the actions and actors who brought about this state is ephemeral.

These problems resulting from the visual and functional juxtaposition of chat and workspace have the consequence that it is hard for users to track and specify relations of content and sequentiality between the textual contributions and the graphical activities. Specifically, there are three major problems:

Deictic references. An important means of communicative expression during collaboration with shared workspaces is deixis [**?**, **?**]–the referencing of objects, relations and actions in the shared visual environment. When chat is used as the communication medium, deictic referencing is associated with high production costs and potentially also higher levels of ambiguity because gestural pointing is not possible. Purely textual descriptions of the object or of its specific position are obvious solutions, but there is no guarantee that such a description will be intelligible to others when they receive it because another user of the shared workspace may have moved or even deleted the object in the meantime.

Decontextualization of actions and messages. When collaborating in a dual interaction space, participants interact with each other through chat messages and modifications to artifacts in the workspace. Whereas the persistent chat history represents the complete sequentiality of the discursive contributions, the same does not hold for the workspace. Both

[2] Despite the fact that this documentation is characterized by incoherence, participants can apparently read and understand the chats amazingly well [**?**].

the ordering and the intermediate results of actions in the shared workspace are fleeting. This has two direct consequences. First, the necessary context for interpreting messages that reference artifacts in the workspace can quickly disappear. This defeats the important advantage of the persistent discourse history, which can support retrospective reflection. Second, the phenomenon of interrupted turn adjacency, described above, is heightened. During the time it takes for one person to respond, others can not only insert new messages but also modify referenced graphical artifacts.

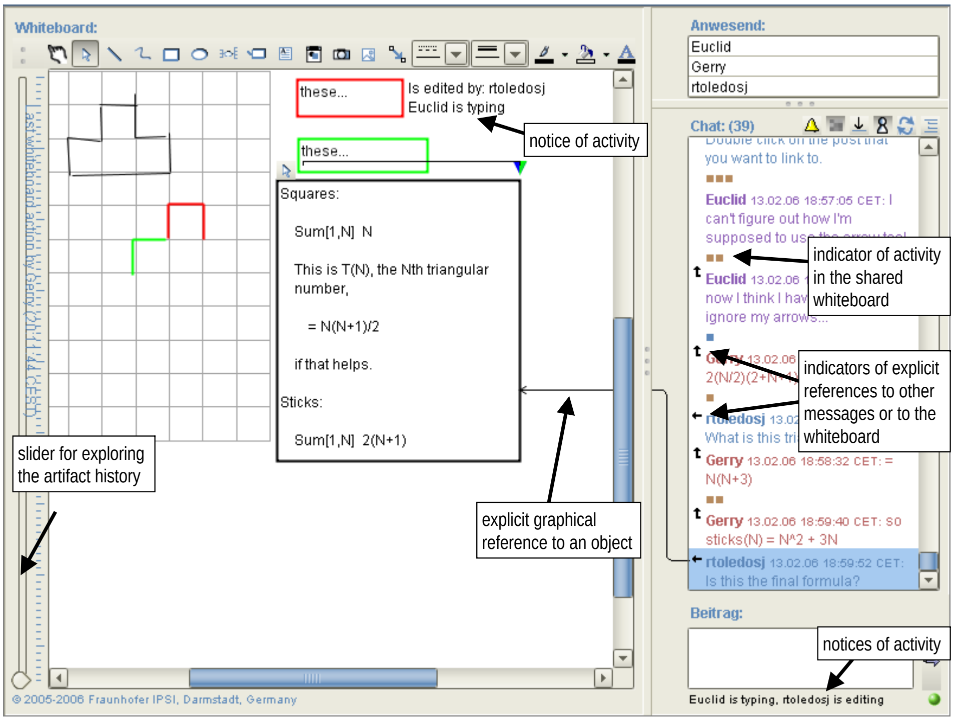
The coordination of communication and interaction. In a dual interaction space, different participants can simultaneously be typing and posting chat messages or producing objects in the workspace. In collaboration, these various activities are interrelated: a message can announce or comment upon an action in the shared workspace and a workspace action can respond to or clarify a chat message. The awareness of the activities of the other people is a prerequisite for the construction of common ground [**?**]. In chat, the chat history documents the sequence of discursive activities of the participants and the usual system messages when someone enters of leaves the room provide basic information about who is present. A series of interface features have been established to support coordination in shared workspaces [**?**], helping with turn taking and the anticipation of actions by other participants. For instance, objects that were just selected by users might be color-coded to indicate who is using them and the location of the user's mouse can be indicated [**?**]. Similarly, many chat systems display a message near the chat input area if someone is typing. However, if all these awareness techniques are combined in an environment with dual interactions spaces, then they can overwhelm the limited attentional abilities of humans. The fleeting awareness messages scattered across the interface require users to pay constant attention to their whole screen.

3 Support through integration

People collaborating in a dual interaction space are exposed to a series of problems that derive from the visually and functionally separated nature of the chat and workspace components. Three software mechanisms will now be presented that integrate these components with each other:

1. An explicit referencing tool that makes possible deictic references from the chat to the workspace.
2. An integrated history function that documents the on-going collaboration process consisting of the activities in the chat and in the shared workspace, and lets users review it.
3. A visually integrated social awareness display that supports the perception of the simultaneous activities of the multiple participants in both areas.

To illustrate these integration measures, a shared whiteboard will be described as a common workspace for the collaborative creation of drawings, concept graphs and mind maps. See Figure 1 for an example showing the most important interface elements.



Whiteboard:
Anwesend:
Chat: (39)
Beitrag:
slider for exploring the artifact history
notice of activity
explicit graphical reference to an object
indicator of activity in the shared whiteboard
indicators of explicit references to other messages or to the whiteboard
notices of activity
Last whiteboard action by Gerry (2h:11:44 CEST)
© 2005-2006 Fraunhofer IPSI, Darmstadt, Germany

these...

Is edited by: rtoledosj
Euclid is typing

these...

Squares:

Sum[1,N] N

This is T(N), the Nth triangular number,

= N(N+1)/2

if that helps.

Sticks:

Sum[1,N] 2(N+1)

Euclid
Gerry
rtoledosj

Double click on the post that you want to link to.

Euclid 13.02.06 18:57:05 CET: I can't figure out how I'm supposed to use the arrow tool

Euclid 13.02.06 … now I think I hav… ignore my arrows...

Gerry 13.02.06 … 2(N/2)(2+N+1)…

rtoledosj 13.02… What is this tri…

Gerry 13.02.06 18:58:32 CET: = N(N+3)

Gerry 13.02.06 18:59:40 CET: so sticks(N) = N^2 + 3N

rtoledosj 13.02.06 18:59:52 CET: Is this the final formula?

Euclid is typing, rtoledosj is editing

Abbildung 1: This screenshot shows the state of the ConcertChat interface after the posting of a message with an explicit reference to a textbox in the shared workspace. Rtoledosj is currently working on the large textbox while Euclid is typing a chat message. The interface features for showing explicit references, the workspace history and awareness messages have been annotated.

Explicit references. The concept of explicit references[3] addresses the difficulty of deictic referencing in the textual medium of chat. Pointing gestures are frequently used in face-to-face conversation [**?**], for instance to identify objects and to clarify relationships among objects. Similarly, explicit references in chat allow one to associate a chat contribution with objects in the shared workspace and with other chat messages using graphical connectors. A graphical reference to a chat message can point to the whole message, a single word or some portion of the message. A reference can also point to an object or a region in the workspace. In the simplest case, one might want to point to a particular object, but in other situations to just a specific part of the object or else to a spatial constellation of several objects. So a number of different forms of referencing must be supported.

For summary statements in the chat-e.g., “These two arguments contradict each other”–multiple references can be made to relevant messages and objects. Just as with gestural pointing, the concrete meaning of a graphical reference may be given only once the verbal message is given. Thus, a reference can be used to clarify a “response-to” relation as well

[3] The presentation of the concept of explicit referencing here is an expansion of [**?**]

as to indicate a “related-to-this-object” relation.

The usability of an explicit referencing tool depends upon its effect on the media-dependent costs of production and reception [**?**]. In order to keep these costs low, appropriate interaction possibilities must be available for the easy production of references and for the visualization of references.

In order to maintain the chronological order of the chat history-rather than threading it-with the associated advantages for retroactive reflection, a reference is represented by a graphical arrow going from the referencing chat message to the referenced object or message. As soon as the referencing message is displayed, the accompanying reference arrow is also displayed, as illustrated in Figure 1.

Artifact history. In collaboration in dual interaction spaces, the actions in the shared workspace and the messages in the chat are but two facets of a single activity. While the chat displays a persistent history of the collaborative discourse, there is no corresponding history display for the workspace, let alone an integrated history for the whole collaboration. In technical terms, an artifact history of the objects in the workspace is a chronological collection of the various different versions or circumstances of the workspace resulting from the manipulations of the participants. In a shared whiteboard, every creation, movement and editing of an object changes the state of the workspace.[4]

The provision of an artifact history has two goals: to preserve the workspace context at various times and to represent its evolutionary process. The context of the workspace at the time when a chat message was being produced is important to know in order to interpret the message-particularly if the message explicitly references artifacts in the workspace. The artifact history permits the reconstruction of that context and encodes that context in the software representation of the reference. As needed, the historical context corresponding to a message of interest can be reconstructed and displayed.

The other goal is to allow the normally fleeting artifact history to be replayed. The chronologically ordered developmental steps can be played back like the frames of a film, making possible reflection on the whole collaborative construction. Reflection in the group discussion is facilitated by the combination of being able to review the past developmental stages of the shared workspace and being able to point to a particular stage with an explicit reference.

Integrated activity awareness. The integration of activity displays has the goal of making it easier to be aware of the simultaneous activity of the other participants. Awareness of these activities is a prerequisite for constructing and maintaining a mutual understanding of the chat messages and the changes to the graphical artifacts–and therefore provides a necessary foundation for collaboration.

In a chat environment, the chat history documents all the activities-both the individual messages and information about participant presence. This chronological documentation of activity suggests that it could serve as a representation of all activity within a dual interaction space as well.

[4] The granularity of the operations depends of the kind of shared workspace. Imagine, for instance the use of a wiki page as a shared material [**?**]. Then, the artifact history would be defined by the various versions of the page.

With chat, the process of producing a message is not directly perceivable by the other participants. The extent to which a long lasting and cognitively strenuous activity in a shared workspace is observable for the other participants depends upon the nature of the workspace and the granularity of the operations that are displayed for everyone. For instance, the editing of a textbox annotation in the shared workspace may only become visible for the others when the edit is completed. Activity awareness notifications have been established to support the coordination of activities like joint editing, so someone knows not to try to edit an object that someone else is currently editing.

In a dual interaction space, however, it is necessary to visually integrate these notices that are associated with the locations of different individual activities. If one participant wants to post a chat message in response to a contribution from another (such as responding to an annotation in the shared workspace with: "I would say that differently"), then she might hold off doing this if she is informed that he has just begun to make a change in the workspace that might very well serve to clarify his original contribution. Conversely, if he is informed that she is typing a chat message, he may delay his change in anticipation of a new objection. Both cases of course presume that the information about the activities is perceived. This can be supported by displaying the awareness information at the appropriate location (see Figure 1).

4 ConcertChat: an example of an integrated interaction space

The integration measures discussed above have been implemented in the ConcertChat[5] system shown in Figure 1. The technical implementation will be described in this section.

The room metaphor. As is common in chat systems, the room metaphor has been adopted here. Participants have to enter the same virtual room in order to communicate and collaborate with each other. Each room includes a chat window and a shared whiteboard, in which the participants can all create and modify geometric figures, pictures, texts and mind maps.[6] The chat and the workspace are persistent. When someone enters a room, the current state of the workspace is displayed for them and the most recent 100 chat messages are loaded. A button is available to load older messages. While only the several most recent messages are visible in the window, one can scroll back to older ones. Each room is assigned a channel for communication with the central server; all information is transferred across this channel: chat messages, workspace operations, awareness messages and system messages.

Message-based client-server architecture. The system implements a centralized controller approach [**?**] to concurrency control in a client-server architecture with partial data replication. Because every action is first sent as a message from the server, the response time (the time it takes for the result of an action to become visible to the actor) is the same as the notification time (the time it takes for the result of an action to become visible to the other participants). This guarantees a unique order to events across the system. The

[5] The system is available online at: `http://chat.ipsi.fraunhofer.de`.

[6] The display of the workspace adheres to the relaxed WYSIWIS approach [**?**].

functionality of the server is restricted to broadcasting to rooms and storing all messages in a database. This results in a high throughput. For message passing, Agilo [?] provides a robust infrastructure–i.e., guaranteed message delivery even in cases of temporary connection failure.

Component-based framework. The main functionality for integrating chat with shared workspaces–explicit graphical references, persistent history of actions, awareness information–is available as a framework for further software development. It provides a set of extension points [?]. A component-based approach allows for the relatively easy integration of various kinds of shared workspaces with the chat, incorporating the different integration functions. For instance, in addition to the shared whiteboard described above, ConcertChat also implements a wiki workspace and a shared picture viewer, in which a user can paste screenshots, picture files or images from the clipboard.

Modeling of the artifact history. Every state of the artifact history can be reconstructed because each operation in the shared workspace was sent by the server as a message to the room before it was rendered by the client, and this operation information is maintained persistently by the server.

Activity model. A replicated activity model exists on the server for every room. Every component that is involved in interactions can store relevant information in this model; the information can then be displayed at an appropriate place in the application.

Modeling and implementation of references. A reference is represented as the connection of a referencing object (a chat message) and a referenced object (such as a rectangle in the shared workspace), together with an additional specification (e.g., the state of a workspace or the starting point and length of a chat message). This description of the reference includes IDs of the included objects, which are resolved by the client upon receipt.

To keep the "production costs" of explicit referencing low for the user in order to increase usability and acceptance, two methods were implemented for creating an explicit reference. The mouse can be used to determine a reference to a chat message or to a region in the shared workspace. It is also possible to select a chat message or a workspace object for referencing with the keyboard.

5 Related work

The concept of the *telepointer* [?] was developed to support gestures during synchronous cooperation in shared work areas. Telepointers are real-time displays of the mouse cursors of all the participants on each other's screens. However, in the context of chat communication these synchronous telepointers are inadequate because the synchronously displayed gestures are not associated with the quasi-synchronous chat messages.

Several systems support an explicit connection within chat to other chat contributions, such as KOLUMBUS [?] or threaded chat [?]. But these systems either do not have workspaces or do not support deictic references to them. Other systems permit ties of chat communication to static documents. In the Anchored Conversations Tool [?], it is possible to create a whole chat thread at any arbitrary point in a document. Explicit references within one contribution to another point in the document is not possible, so that the different chat discourses at different points in the document have no connection to each other, leading to the danger of discourse fragmentation similar to that in the threaded chat approach. Similarly, in Kükäkükä [**?**], web pages can each be combined with a threaded chat, but with no support for deictic references to specific parts of the page.

An alternative approach is followed by systems like CURE [**?**], in which individual artifacts (here, mathematical formulae) can be included in a chat message. However, these artifacts cannot be collaboratively constructed or manipulated. The GraffDis-System [**?**] takes the opposite approach, integrating textual communication into the workspace. Participants can create contributions of text, graphics or other materials at any point in a shared workspace. But these contributions disappear over time. The chronological sequence of the shared history can be navigated with a *history slider*. While the ability to position contributions anywhere permits the expression of connections, this is only possible during a restricted time window because the older contributions fade away. Just as with threaded chat, this approach also entails the danger of fragmenting the discourse.

None of these systems for synchronous collaboration address all three aspects of the integration of chat communication with a shared workspace.

6 Experiences and future work

Since 2004, the Virtual Math Teams Project[7] has been using a collaboration environment based on ConcertChat for the discussion and solution of mathematical problems by small groups of students. A detailed case study of how deictic referencing was conducted in this context using the ConcertChat functionality in the dual interaction space is presented in [**?**]. Further studies of the use of ConcertChat's explicit referencing tool are reported in [**?**]. These provide some evidence that the participants were able to employ effective communication strategies with the help of the explicit referencing.

For researchers, the persistence of all activities in a dual interaction space provides the possibility of conducting fine-grained analyses of group interaction, as illustrated in [**?**]. To support this, a replay version of ConcertChat has been developed that allows all the activities to be repeatedly reviewed, with the chat and workspace histories precisely coordinated.

The experiences with ConcertChat to date suggest a series of further research questions.

1. The storing of explicit references and the integrated representation of all activities make available additional structural and temporal information about the collaborative artifacts in the two interaction spaces. To what extent is it possible to use this information to construct a retrospective documentation of the collaboration that would facilitate future reflection or recall by the participants-for instance, when they return to the room for a subsequent session?

[7] The Virtual Math Teams project is available online at: http://mathforum.org/vmt.

2. A dual interaction space is intentionally designed to provide a variety of modes of working. A chat provides the ability to type new messages at the end of a sequential discourse. A shared whiteboard lets users create, edit and arrange graphical elements using direct manipulation. Consideration of such issues raises the question, which forms of objects and what functionality are optimal or appropriate for specific collaborative tasks, media competencies and educational goals?

3. An essential difference between a chat window and a shared whiteboard is the persistence of the artifacts [**?**]. While a textbox in a shared whiteboard remains visible indefinitely (unless it is edited or deleted by a participant), the same is not true for chat contributions; they scroll out of sight with the appearance of the following discourse. Interesting questions arise when the additional possibility of audio communication offers a non-persistent medium. Can this supplementary mode of communication be substituted for chat to the advantage of the participants or will it be used as a secondary addition? What different communication strategies would result?