Message capturing as a paradigm for asynchronous digital whiteboard interaction

Lutz Gericke, Raja Gumienny, Christoph Meinel
Hasso Plattner Institute Potsdam
Prof. Dr. Helmert Str. 2-3, Potsdam, Germany
Email: {firstname.lastname}@hpi.uni-potsdam.de

Abstract—Remote Collaboration in synchronous and in asynchronous communication settings demands for highly specialized solutions. The requirements are set even higher when teams are working creatively with methodologies such as Design Thinking where challenging and unusual problems are addressed. Based on our first *Tele-Board*[1] prototype which supports real-time Design Thinking for teams distributed over different locations we implemented extended functionalities for asynchronous interaction support. The "Tele-Board history browser" is a web-based user interface offering functionality to go back and forth in the timeline of a whiteboard. Additionally it is possible to view the whiteboard's usage statistics to gain insights how Design Thinkers work. In this paper we describe the system's architecture and the role of message capturing as an efficient way for saving and displaying a whiteboard history as well as for profound research data collection. With our tool we can support designers in fulfilling their common tasks more efficiently in dispersed teams and we can also assist design researchers to understand how designers work in an all-digital setting.

I. INTRODUCTION

Collaboration over distances can be facilitated by the use of computer supported collaborative work (CSCW) systems. With the help of these tools, teams who are located at different continents can share their work and communicate despite of large distances between them. To enable communication being comparable to co-located work settings people usually use real-time communication systems, which include video conferencing or shared desktop applications (cf. II). However, working across continents often involves huge time differences and therefore teams can only work synchronously for a few hours – if at all. This problem results in less joint work and leads to more reporting and documentation via e-mails and other documents about the efforts at each location. As we learned from interviews with employees of huge global companies documentation is time-consuming and frustrating for all involved parties and only done because project management demands it.

Besides these difficulties, e-mails and text documents often are not suitable to convey ideas and concepts that people have during their project work. It is very hard to communicate and understand why people took certain decisions and which were their most important concerns. Especially with creative work, which involves a lot of unforeseen ways of working, thinking about ideas or innovations and visualizations of concept and designs, it is very difficult to write down the results of a meeting. Teams who are applying methods such as Design Thinking [4] often work with whiteboards, sketches and sticky notes and only in the end they "translate" their work to text documents or presentations.

To address the problems of Design Thinking teams who are working asynchronously over distances, we developed the *Tele-Board history browser*: a web-based interface giving the opportunity to go back and forth in the timeline of a digital whiteboard. It enables the design thinker to view the gathered data from different perspectives and thereby gain a deeper understanding of the project context [27]. Additionally it supports the team to analyze the overall project progress and decision paths taken by the respective distributed subteam or by the team itself in an earlier project phase. The team can also continue at any past state by duplicating the whiteboard content, i.e. starting a parallel session. All data is saved implicitly, meaning that the user does not have to care about it.

Furthermore it is possible to view the whiteboard's usage statistic to gain insights how the designers work. Important areas on the whiteboard (hot spots), time periods with a lot of interaction or different project phases can easily be detected with the collected data.

In this paper we describe how our digital whiteboard system works and how message capturing can be used to store interactions (cf. IV). We present our corresponding applications for user interaction and show how valuable the collected data can be for analyzing design work and possible statistics (cf. V).

II. RELATED WORK

A variety of computer supported collaborative work (CSCW) systems have been developed in the last decades to facilitate work over distances. These systems often focus on only one or two of the fields in the commonly known Time / Space Groupware Matrix (see Fig. 1) [7], [18], [25]. To support synchronous work, most existing solutions concentrate on either video conferencing or desktop sharing [9], [14]. A frequently used tool that offers both functions is Adobe Acrobat Connect Pro [3], a web-based conferencing system and so-called learning environment. It features the most common tasks in a meeting setup including audio and video conferencing, screen sharing and a simple whiteboard solution. But the integration between these components is insufficient.

[1] http://tele-board.de

For example, pointing at certain parts of a sketch on a whiteboard is impossible in a video conference. Interviews with employees of a large software company who are using the software intensively showed that most of the functionality (e.g. the whiteboard component) is hardly used.
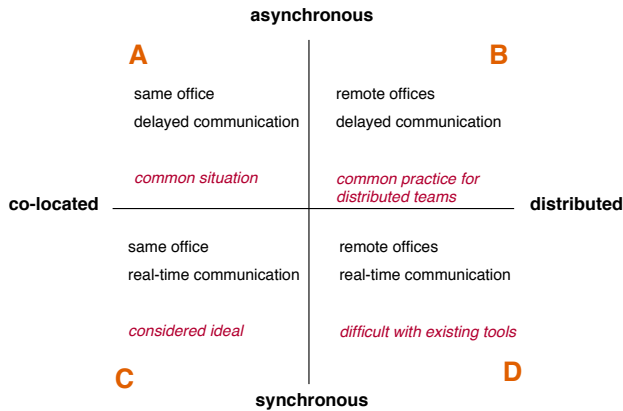

Fig. 1. Categorization matrix of working modes in CSCW

Telepresence systems such as those provided by Cisco [6] or Polycom [22] are elaborated high-end video conferencing systems. High definition video and audio as well as special security features make it only affordable and best suitable for big companies. Telepresence systems are basically an arrangement of hardware components. The most elaborated setup makes it possible to build up a virtual meeting room to give everyone in the meeting the illusion of sitting together at the same table. The critical drawback for creative work is the missing support of synchronous whiteboard interaction.

There are several commercial and non-commercial web applications which focus on enabling the user to sketch ideas on whiteboards (cf. [1], [2], [12]). All of them provide functions to draw sketches and share them with colleagues. Real-time collaboration on the whiteboard is not possible and would be difficult to implement, because none of these solutions offer support for audio or video conferencing. There are also several research projects which concentrate their work on whiteboard systems and usage (e.g. [8], [15], [17], [28]–[30]). Some research prototypes combined a shared (whiteboard) workplace with a video, e.g. *VideoWhiteboard* [29], *Clearboard* [17] or the augmented *Designer's Outpost* [8].

All of these tools concentrate on synchronous working and do not take into account that users work at different times in different time zones. Some of them do offer a save and load option of a whiteboard state, but it is implemented in the traditional way of file administration. This includes that you have to care about saving and loading and can not leave content on a whiteboard as you would do with a none digital one [27]. Klemmer et al. first implemented the possibility to go back to different states and even try out parallel interactions from a certain (decision) point within the whiteboard's timeline [19]. However, the authors point out, it is sometimes problematic to reconstruct certain whiteboard

states as their system is based on paper sticky notes (which are partly digitalized) and degenerated states can occur.

Furthermore, there are existing applications using tracing of past activities, e.g. source code versioning or document management. But capturing whiteboard data and storing arbitrary versions of it, has not been done before.

In consideration of all interviews and observations of existing tool usage we believe that an efficient support can only be guaranteed if the approach encompasses all dimensions of collaboration, i.e. synchronous and asynchronous as well as co-located and remote settings. In the following section we will describe the requirements for such a system and our previous implementation.

III. REQUIREMENTS FOR ASYNCHRONOUS WORK

In earlier research work we developed a fully-functional whiteboard client prototype, which enables people to have a synchronous whiteboard interaction with an underlying video conference [13]. People can work together in a digital design space, sketch and develop ideas and interact at one location as well as between distributed locations. It is possible to write sticky notes with a variety of different input devices and write or draw on whiteboards with pens. All interactions at each each whiteboard are synchronized via a server to guarantee the same view on all artifacts. The focus in this previous work was set to a very natural design experience in order to supply people with the tools, they are used to from traditional whiteboard environments. From observations we encountered some core working modes and managed to support them with our prototype.

Nevertheless we also observed and heard from interviews that asynchronous work is sometimes even more common than synchronous work as it is difficult to find a shared time slot for a meeting – even for teams at one location. In practice people often start working on a problem and their colleagues continue at another point in time [27]. To understand what the others were doing and when they made certain decisions is crucial for team interaction and a common ground of all involved parties. Preferably, the teams should not be bothered with complicated saving and loading of their work as this is unnatural with regard to the traditional whiteboard metaphor and probably would not be used. It must be easy to navigate through different whiteboard states and go on working at any previous point in time.

A digital whiteboard solution could also offer the possibility of extensive and partly automated documentation. In traditional whiteboard settings it is time-consuming and troublesome to take detailed photographs after work is done. Written documentation for stakeholders and customers has to be prepared additionally. If this step was unnecessary, a lot of time could be saved and dedicated to more creative work.

Another aspect for the importance of an implicit documentation is the statistical relevance for design researchers. Various questions could be answered using the history data: What is the main working time of the employees? How can the produced results become measurable? How is activity distributed in team

setups? Not only design researchers could be interested in this information, but also the designers themselves would profit from gaining insights into key factors of their creative work.

In the following chapter we describe how we addressed these requirements and outline the system's architecture.

IV. DESIGN OF THE SOFTWARE SYSTEM

To address the mentioned challenges and to realize the functionality in a software system we developed three main functional units:

- interception of message flow
- storage of communication data
- enabling interaction with the history data in an appropriate user interface

The communication should be captured on-the-fly, which has influenced the selection of the technology insofar as it must be possible to analyze packets separated from the message routing. Central roles in the overall system are represented by the message server and its plugin architecture, the web-based management system, and the database management system. The history functionality is a concept that is implemented as a cross-cutting concern in all parts of the system. It can not be realized as one single component, because it enriches the functionality of the other components. A client-server architecture is used for synchronizing the participating whiteboards. A central history archive located at the server is more suitable than e.g. a peer-to-peer model as it is used in [10], because all statistical data should be kept together in order to be analyzed conveniently and enable asynchronous work.
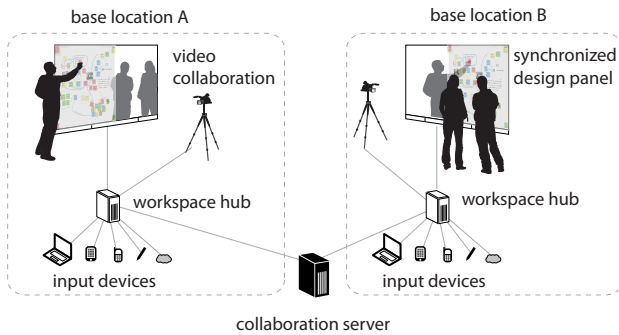

Fig. 2. Overall setup of the communication infrastructure centered around the workspace hubs that are connected via a communication server

To understand where the communication takes place and what should be captured, the existing communication infrastructure should be briefly outlined: A *Workspace Hub* is a computer system running the whiteboard client software and is located at a digital design space. This computer combines all physical components at this location. Several input devices are paired to it. Cameras make it possible to also synchronize a video stream of the people standing in front of the whiteboards in order to provide a communication channel supporting eye-contact and gestures regarding the whiteboard elements (e.g. pointing on a sticky note). How the multiple Workspace Hubs

work together and how they communicate with the help of the server is shown in Fig. 2.

Naturally, the best place for capturing communication between the whiteboard clients is the communication server. An interceptor was realized as a plugin in this server (see Fig. 3). In our operational prototype, Openfire [24] is used as the server software, because of its extensible architecture. The so-called *Server-Buddy* plugin thereby captures all incoming packets and stores them in a database. Special packets such as a request-message for resuming a session are filtered out, directly answered and not stored in the whiteboard history.
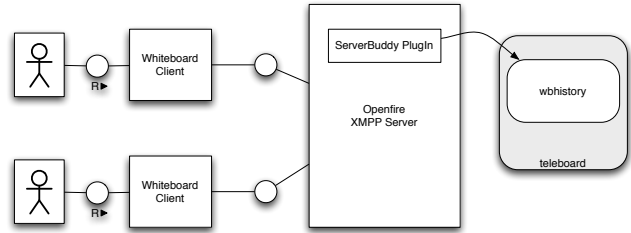

Fig. 3. Communication interception between the whiteboard clients

Projects and *Panels* are important concepts in this context. A Panel p describes the sequence of events e_n executed on one whiteboard in temporal order of these events ($p = (e_1, e_2, e_3, ...)$). An event is a tuple of attributes describing which action has happened where, by whom, and when, to keep the temporal order of the events. Each event has an operation code, which can be NEW, $CHANGE$ or $DELETE$ to describe the event type. A Project pro is the collection of multiple Panels ($pro = \{p_1, p_2, ...\}$) in order to configure them with rights to edit/view/delete.

A Panel can be thought of as one whiteboard session with the timeline of events and actions that have been made on it. The typical history sequence of a Panel could consist of: sketching, placing and moving sticky notes, erasing of single items, but also the panning of the whiteboard. So in general an event represents a state of a whiteboard element, but can also express an action executed on the whiteboard that has no direct influence on the whiteboard content itself, such as scrolling the surface.

A. User interaction points

Basically there are three major classes of users interacting with the developed system: the designer working on the whiteboard content, a manager tracing the design activity of the designers, and a design researcher who wants to gather insights on how the designers worked. The first two types care more about *what* has happened, the last one addresses more *how* the interaction took place - of course there are large overlapping parts in these activities.

Fig. 4 shows these components that are interesting for understanding the interaction process. Users have three major points of interaction with the system. The first one is the web-based management system. The user logs into the web portal

and browses through Projects and Panels in order to find a preview of the past design sessions.
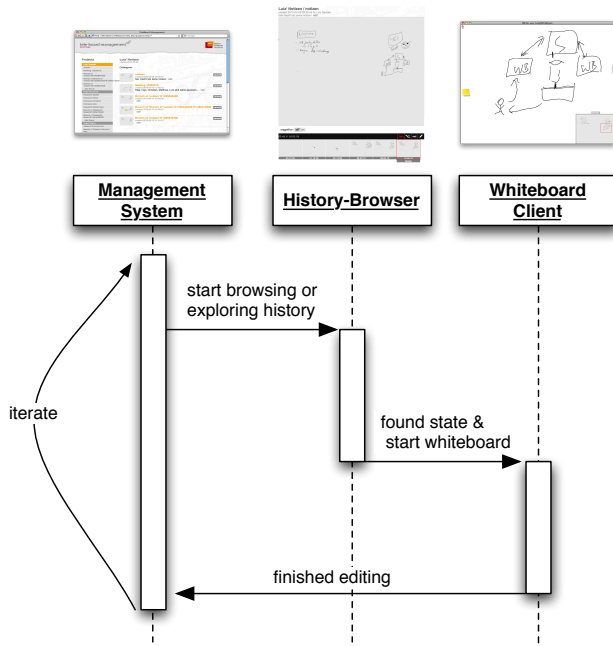

Fig. 4. Sequence flow between different application components

Embedded in the web-interface is the *History-Browser*: a user interface enabling people to go back in the history of a Panel. A user can immediately see changes between points in time by switching between the screenshot images of the whole whiteboard. By skipping through days, hours, minutes or even seconds, differences can be found immediately. If you search for a certain series of events you can zoom into the history of the Panel. At each zoom level, the user only sees those time periods where action took place. So e.g. if you zoomed into the 11th hour of a certain day and only in the first 20 minutes there were events on the whiteboard, only the first and second 10 minute time slots are shown. A time-continuous approach was prototyped earlier and evaluated as not appropriate because too many unimportant time periods would be shown.

The third component is the *Whiteboard Client*. It is the main component, since design teams will spend most of the time with this interface. The standard configuration uses a touchscreen device with a whiteboard form factor, such as a SmartBoard [26], to interact with the software. You have the possibility to draw sketches, place sticky notes on the board, cluster objects to rearrange or only group them visually. As all events are synchronized, teams can work simultaneously, which is supported by a video conferencing system in the background. People interacting with the board see each other through the transparent digital whiteboard screen, as if they wrote on the same wall and stood next to each other. The use of gestures as well as face-to-face communication is possible with this setup.

When a Whiteboard Client is started, it connects to the server passing a certain Panel identifier. The latest state of the whiteboard content in this Panel is requested from the server and sent back to the client. From that state, people can continue working on the selected Panel by editing the whiteboard content. Each update is sent to the server which redirects messages to the so-called "partner-whiteboard" - the paired remote whiteboard. Thereby it is ensured that people on both locations see the same content.

Another possibility for the user to interact with the system at the current location is using one of multiple possible peripherals. A smart-phone for instance can be used to write sticky notes, either by sketching graphically or typing text on the keyboard. The used protocol supports sending of text-messages from many devices to the whiteboard client which then are displayed as text-based sticky notes. Digital cameras can be used to transfer image files to the whiteboard. There is also a special client software that can be used on a Tablet PC to create sketched sticky notes.

Besides the introduced user interfaces, there is a set of background services, providing functionality that can be used in the user interfaces. The most prominent feature is the screenshot generation. The database content that is recorded by the history, can be used to reconstruct a point in time of the Panel. The resulting images are used for several purposes, e.g. the History-Browser as well as in an overview map in the whiteboard client. Other background services enrich the user experience by providing visualizations and analyses of the collected data.

B. Communication protocol

There is a large variety of input devices and thus highly different requirements for the communication protocol concerning the connected devices and the whiteboard client. Generally there is a differentiation between Tele-Board-aware devices being equipped with a special client software and those coming out of the box that can participate in the design session with an existing client software that uses the communication protocol. It should be possible to write and send sticky notes from every Internet-enabled device without developing a special client software for it. When using special capabilities such as drawing it can be acceptable to develop an adapted application for that platform.

Every single component has very special needs in terms of user interface development, data structures, and communication methods. So an important decision was using the Extensible Messaging and Presence Protocol (XMPP) [21] as a communication protocol as a consequence of the variety of input devices and the different platforms that should be supported. There are several implementations on almost every platform and it is supported by a large variety of existing clients.

XMPP is an open standard and is typically used as a chat and instant messaging protocol. Over time it has been extended to support voice, video, and file transfer. XMPP (formerly known as Jabber) is used in several instant messaging tools such as Google Talk [11] or Psi [23]. The communication is build upon a client-server model. Authorization, session

and roster handling is managed by the server. People can connect with every possible client without transferring any configuration from client to client except for username and password.

The Server-Buddy plugin, which is deployed into the Openfire server, acts as a so-called PacketInterceptor to read all messages sent between whiteboards in order to archive them in a database. This procedure makes the collected information usable in the history component, so that every state of a Panel can easily be reconstructed.

Technically, all communication is routed over the XMPP server. In terms of XMPP, the whiteboards talk to each other. XMPP-Clients producing text-based sticky notes, direct their messages to a specific whiteboard, so that the physical location of these two components is reflected in that logical model. The used method for communication between any two partners is a simple chat, as it is defined in the XMPP specification. The body of the messages is extended to support the needs of the Tele-Board synchronization.

There is an operation code signaling the kind of message directly stored in the message body. There are multiple types of operation codes (opcodes) used:

- `WHITEBOARD_SYNC_NEW` - an element is newly created
- `WHITEBOARD_SYNC_CHANGE` - an element is changed
- `WHITEBOARD_SYNC_DELETE` - an element is deleted
- `WHITEBOARD_SYNC_CLEAR` - the complete whiteboard is erased
- `WHITEBOARD_SYNC_ALL` - request to send all whiteboard content to the communication partner
- `WHITEBOARD_SYNC_ALL_ANSWER` - answer to a sync-all request, contains the whiteboard content
- `WHITEBOARD_SYNC_ANNOTATION` - sends annotation to the server

There are multiple devices with different capabilities concerning the handling of the XMPP messages. Messages using the mentioned opcodes are sent between the whiteboard clients to synchronize the whiteboard content. The peripherals will send the message content to the local whiteboard. From the Whiteboard Client the content will be synchronized to the remote whiteboard location. The `WHITEBOARD_SYNC_*`-messages are important for the whiteboard history and will be handled in the Server-Buddy plugin to build up the historical data. `NEW`, `CHANGE`, `DELETE` will be directly archived, `ALL` and `ALL_ANSWER` will not appear in the history and `ANNOTATION` will be stored in a separate database.

XMPP as a communication language between the clients turned out to be very appropriate. The development of the whiteboard clients can rely on a sophisticated infrastructure e.g. for user handling and message routing. It did not have to be implemented from scratch, but could be used as an existing part of the protocol.

The payload of the chat messages between the Whiteboard Clients is an XML-encoded text representation of a single whiteboard element. The example in Fig. 5 shows a set of Skribbles, having different properties; the XML extract, shows one single Skribble element. There are several attributes describing the element in order to reproduce it at the remote location: `x` and `y` describe the position, `strokecolor` the color of the path, and `d` represents the path itself in SVG-notation. SVG was chosen because it is an established standard and can be directly used for screenshot rendering with only little string conversions as well.
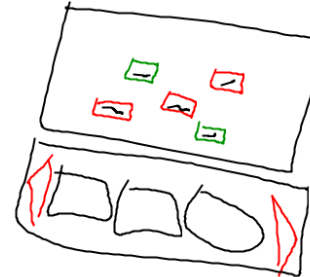

```
<path id="lutz@fb10dtools_654"
    strokecolor="0.0,0.0,0.0"
    d="M 2286.0 1237.6575 L 2283.0..."
    x="119.0" y="354.0"/>
```

Fig. 5. XML representation of one single Skribble vs. graphical representation of multiple Skribbles with different colors, paths, and locations

C. History-Browser

The History-Browser is the tool, which enables the user to browse through the archived whiteboard data. It is possible to go back in the timeline of interaction and reproduce every point in time of a collaboration session. This application is visually part of the portal interface, but logically separated from it. It goes a step further than the Tele-Board whiteboard client and extends its functionality. The aim of the application is to show how a Panel has developed over time. It offers a read-only view of the whiteboard content and offers an entry point into existing whiteboard sessions.

The current version of the History-Browser is the result of an iterative development process. The first versions used a time-synchronous approach to display the history data. It turned out not to meet the user expectations, e.g. when working on a Panel for an hour on one day and a week later for another hour, the timeframe where there was no interaction at all takes most of the screen area in the user interface. We decided to only show those periods with archived interaction to make navigating more convenient.

One main problem during the realization of the user interface was the permanent shortage of screen space. The whiteboard client can use the whole screen for displaying only a region of the whiteboard. With the History-Browser (see Fig. 6) it should be possible to see the whole whiteboard surface (which is about 5x5 times of the usual screen's size) in one browser window as a whole. An in-browser magnifier tool makes it possible to highlight certain areas of the downscaled whiteboard surface revealing its original dimensions to make it easier to read text on sticky notes.

Fig. 6.   History Browser user interface embedded in the Tele-Board portal

There are several temporal zoom levels for adjusting the amount of detail that is shown. The user can switch between the units of days, hours, ten minutes, minutes, ten seconds, and even single seconds. Fig. 6 shows an hourly partitioning.

Screenshot generation: The screenshot generation is one of the most crucial parts of the history functionality. Goal of this functional unit is generating whiteboard content out of the database into a graphical representation. In several parts of the whole Tele-Board system, there is a need for a fast and convenient screenshot generation. In the portal, every Panel has a preview image and of course the History-Browser makes extensive use of the screenshot generation. For every zoom level, about 10 images have to be rendered as thumbnails and one image for the larger preview area. The whiteboard client uses these screenshot images to display an up-to-date overview map. Theoretically, the applications could be realized by a client such as the whiteboard client, receiving the historical data via XMPP, but performance tests showed that the XMPP-transfer is not as fast as the direct database connection that can be used via the PHP MySQL interface. Furthermore, a thumbnail PNG image is by far smaller than the XML-based representation in an XMPP message. These images can also be much easier used on webpages or in e-mails.

The general procedure for image generation passes through three steps:

1) extraction of whiteboard state out of the database
2) conversion of database content into SVG representation
3) optional: conversion of SVG file to PNG format (raster-ization)

The protocol that has been described for whiteboard synchronization communication is based on XML and in some parts very similar to SVG, e.g. the Skribble path notation matches exactly the SVG notation for paths. This reduces transformation complexity in step 2 when generating SVG from the whiteboard content. For special purposes, the SVG image can be directly transferred to the user, but in many cases this will not be the preferred solution. SVG files of a full whiteboard can be tens of Megabytes in size. If only a small preview image is needed, a PNG file of only a few kilobytes will be far more convenient.

Measurements revealed that the computation time used for the conversion from SVG to PNG is the biggest portion of time consumption for creating the images. Therefore, it is immensely important to optimize this process step. There are several software packages available to do exactly this rasterization computation. Comparisons showed inkscape [16] - an open source vector graphics toolkit - to be the best performing tool in our context. But this conversion can still take up to multiple seconds. Therefore we introduced caching routines in order to minimize the SVG2PNG conversion requests. With cache hit ratios of over 75% for PNG file generation and over 90% for the SVG generation task, the implemented caching strategy turned out to be very valuable.

D. Asynchronous history-based interaction

The History-Browser enables users to navigate to a certain point in time in order to understand the collaboration process that has been conducted using the Tele-Board system. When the user has navigated to an interesting state, currently two options are offered: resume or branch a Panel and comment or email an interesting point.

The central concept behind the history archive is the semantics of an implicit save and load behavior. Because every single whiteboard interaction is captured in the system, every session can be resumed at every point in time. This is done automatically, no work will be lost by unpredictable interruptions.

From a technical viewpoint, resuming is only possible from the end of a whiteboard session, because everything that is recorded lies in the past and cannot be modified without changing the ongoing events. In the user interface this limitation is not existing anymore. Users can resume virtually every whiteboard state.

Resuming at the end of a session would be simply starting the Whiteboard Client, transferring the latest state from the server to the client, and beginning the typical whiteboard activity. Continuing a session from any point between start and end leads to a branch in the back-end. Branching means, that the selected state would be copied into a new Panel, which than can be resumed. Branches are typically used to try out ideas in a parallel session without influencing the ongoing main design session.

Branching now makes it possible to resume a state at any time of a Panel. This is exactly what a user would do when he branches in a classical analog whiteboard: he erases everything that is not needed in the ongoing session. This analog behavior has two major drawbacks: the original copy does not exist anymore (maybe except for photographs taken beforehand)

and it is impossible to branch from a state that was already erased. These problems can be overcome with the digital solution.

Large parts of the underlying computations are realized using a database management system using standard SQL. The most crucial query is the generation of a whiteboard state. An example of the history database table is shown in Fig. 7.

id	create_time	panelid	opcode	obj_data
2	18:13:45	0	NEW	<path id="wb1@fb10dtools_2" strokecolor="0.0,0.0,0.0" d=" M 3411.0 2536.0" x="3411.0" y="2536.0"/>
2	18:13:46	0	CHANGE	<path id="wb1@fb10dtools_2" strokecolor="0.0,0.0,0.0" d=" M 3411.0 2536.0 L 3409.0 2533.0" x="3411.0" y="2536.0"/>
2	18:13:53	0	CHANGE	<path id="wb1@fb10dtools_2" strokecolor="0.0,0.0,0.0" d=" M 3411.0 2536.0 L 3409.0 2533.0 L 3398.0 2520.0" x="3411.0" y="2536.0"/>
2	18:13:57	0	DELETE	<path id="wb1@fb10dtools_2" strokecolor="0.0,0.0,0.0" d=" M 3411.0 2536.0 L 3409.0 2533.0 L 3398.0 2520.0" x="3411.0" y="2536.0"/>
..	..	..	..	..

Fig. 7. Excerpt from whiteboard history table "wbhistory"

Querying a whiteboard state, can become costly in terms of computation time, because the sequence of events of every single whiteboard element must be analyzed for this reconstruction process of a Panel. The typical element lifecycle will follow the order NEW → CHANGE → DELETE, while there is a potentially very large number of CHANGE events. When now querying a state, it is necessary to retrieve only the latest element, because in every `obj_data` attribute, the complete representation of a whiteboard element is stored. If the latest element is a DELETE operation, this event can be removed from the result set, because the element is not existing on the Panel anymore. Delta representation of a whiteboard change event would lead to immense costs for state reconstruction and therefore was not used.

When the whiteboard client resumes a Panel, it connects to the communication server first, sends a `WHITEBOARD_SYNC_ALL` message to the server, and than Server-Buddy answers via XMPP with the latest whiteboard content. Earlier versions of the whiteboard client could reply directly to that message and send the current whiteboard content to the partner. With the current solution the Server-Buddy can now replay this latest whiteboard state without the existence of a partner whiteboard client. This also allows asynchronous co-located work, which is also stored in the history archive (in most cases co-located work is actually asynchronous). When the Server-Buddy answers the request, the request message will be dropped, so that no further whiteboard client will receive the message and accidentally send a second response.

The History-Browser is a very valuable tool, when finding certain points in time and retracing activity on a very detailed level. This is especially useful when a user browses the history who was already participating in the design session. Besides this interactive solution the History-Browser provides, we developed an additional application that renders a movie from a series of screenshots. The still images are taken from every point in time, when action took place. Thus, every interaction can be seen in the video. This movie playback can be a more passive way of exploring the history towards an overall understanding of an unknown session that leads to exploring the history more detailed and with a better temporal classification of the content than using the History-Browser.

E. Performance considerations

The used database table is a very simple way of storing the large amounts of data. However, it turned out to be perfectly suited for the reconstruction of a whiteboard state. No XML parsing is needed to generate the whole whiteboard content. A string can be produced by concatenating all `obj_data` attributes into one large, XML-compliant string.

The data model turned out to be very adequate. Normalization of the database schema would not result in a performance gain for most queries. The costs spent for storing data in the database must be as low as possible, because these requests block the real time communication system. The insert times for the current solution are minimal; first on the database level, but also in the Server-Buddy, which has to spend only little time on parsing and analyzing the received packet.

Over time, the collected messages accumulate in the database. For performance as well as clarity reasons, we have a semi-automatic way of archiving unused data as it is and store it in a separate database. On demand, users can recover the data into the live system. In a future version this archiving could be done automatically for infrequently used whiteboard sessions.

To answer statistical questions it would be more suitable to extract every XML attribute into one column of a new table. This makes it possible, to answer questions such as "What regions of the whiteboard are used most?" instantly without parsing XML data during the query. To minimize the time-consumption for XML parsing, an additional periodic transformation process was introduced. The XML element data is parsed once and the key/value-pairs are written into a new evaluation log. This job typically runs once a day, which is completely sufficient, because live statistics data is not need in most cases. The statistical insights that can be derived from this evaluation log will be shown in the next chapter.

V. History data as a basis for analyzing designers' interaction

A typical design research question is "How do designers work?". To answer this question, continuous observation is needed towards an deeper understanding of their activities. Previous approaches such as iLoft [20] or the *Design Observatory* [5] use elaborated techniques and tools to capture the behavior in the room where the observed designers work. One point that is mentioned is that these observation instruments can lead to distraction of the design activity. By implementing

the observation instruments as a part of the tools they actually use, this distraction will be eliminated and even the designers can benefit from the digital enhancements, such as resuming historical sessions.

One major drawback of the previously mentioned approaches is that they only capture an image stream of the interactions. There is only few context information available and large effort has to be spent on manual analysis of the raw material. With an all-digital solution, this process can be automated for the most part. It also offers the possibility for immediate feedback, because analyses take less time. This can lead to faster iteration cycles for experiment setups and better results in the end.

The designer can also benefit from statistics of his work. With a distant view on the past work, designers can replay what has happened and come to further insights for their future work. It also enables the participants in asynchronous settings to better evaluate their personal contribution design task and also better value the work of their partner team.

The statistical analyses that are shown in the following can only give an impression of the potential the history archive is able to provide. More elaborated statistics can be created when ongoing test series are completed. Structural connections between certain Panels can be seen intuitively, but have to be refined to fulfill strong statistical criteria. The preliminary results therefore outline what kind of statistics will be possible in the future. These data can be used for several applications such as searching. By analyzing the processes, important points of the whiteboard content could be made easily accessible. This definition and extraction of outstanding phases in the sessions could also lead to a compact report for an asynchronously working colleague by automatically giving hints on which points in time would be more valuable than others.

A. Spatial analysis

One of the most intuitive ways of analyzing the whiteboard content distribution is a map of the content. Fig. 8 shows three examples of a spatial analysis. You can see that there is a distinct concentration of activity at one spot on the whiteboard surface, showing high activity e.g. caused by creating and moving sticky notes or the creation of sketches.

There are also many other ways of visualizing the content. To get a deeper understanding of the scrolling behavior of the people using the whiteboard client the chart in Fig. 9 has been created. Scrolling of the whiteboard surface is realized in the client as a wipe gesture as you know it from most touchscreen applications. An interesting point for us to see was how people used this function. Typically they use the default scroll area of the whiteboard and when it is full of drawings and sticky notes they scroll sideways until they have a whole new and empty whiteboard. This behavior can be clearly visualized with the shown chart by a characteristic wave pattern. This insight influenced the whiteboard client development insofar that in future versions the scrolling controls will be adjusted to better support this typical behavior.
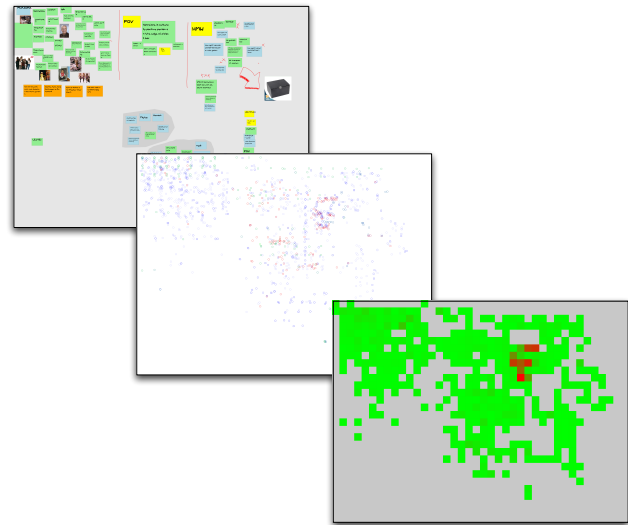

Fig. 8. Spatial usage map comparison, from left to right: whiteboard content screenshot, map using circles to denote events, heatmap displaying a two-dimensional histogram of whiteboard event activity (red to green = high to low activity)
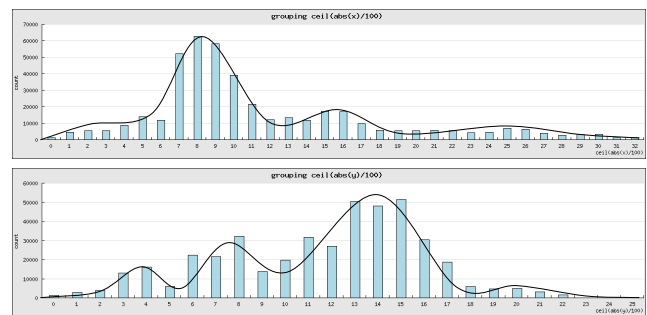

Fig. 9. Histogram chart of whiteboard activity grouped into segments of 100 pixels, either in only horizontal (top) or vertical direction (bottom)

B. Temporal analysis

The temporal activity distribution is also a very important dimension in terms of historical analysis of the archived data. It can give insights about the project lifecycle and when participants were active. A chart can be generated as a histogram of the working hours, not only in general but also for every participant. This gives the possibility to estimate if working time was used efficiently, but is not supposed to give any qualitative feedback.

Another visualization that is encouraging for further research is shown in Fig. 10. You can see an overview of all whiteboard elements that have been created within a given Panel. From top to bottom, it is the sequence of distinct elements in the order of their first appearance. The horizontal dimension shows the time axis. Horizontal lines show the lifetime of a single element, whereas a red line describes a sticky note and a blue line a skribble. Blue dots show change events on elements. As you can see, there are patterns identifiable. Certain phases in the design process are reflected

by blocks in the graph. The data that is shown in this figure is taken from an experiment at the School of Design Thinking in Potsdam, Germany. The participants had the task to synthesize their user research insights for a given problem question.
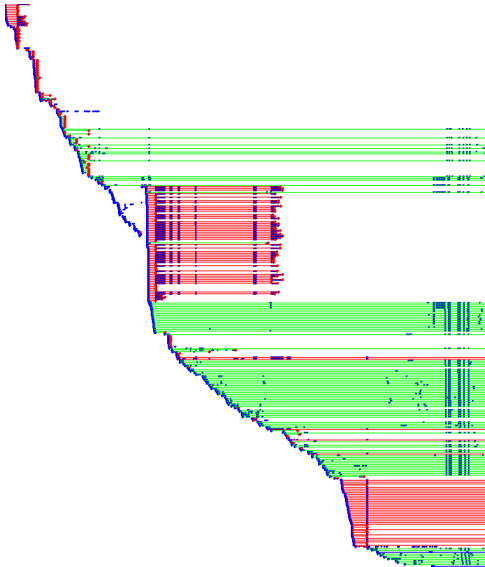

Fig. 10.   Whiteboard session development including the lifeline of every whiteboard element

These usage patterns are recognizable also in other whiteboard sessions. This kind of analysis can be used to enrich applications building upon that data, such as a search interface or context-aware assistance that shows special tools by analyzing the current working mode. Summing up, especially the temporal analysis can give large insights on how the design teams worked.

C. Key figures

Besides the mentioned visual ways of analyzing the whiteboard content, there are other key figures describing the interaction processes. One example is the reasoning on the number of sticky notes that have been either made smaller or larger. From these numbers an argumentation can lead to optimized sticky note default sizes for different tasks.

Very important is also to study the contribution of activity by each team member in a distributed setting. Therefore we set up an experiment with 10 teams (with two participants each) being located at two different digital design spaces using the Tele-Board system and having different configuration parameters (e.g. audio-only vs. video-enabled setup). The contribution of each participant differs very much. This is also reflected in the history data and can be analyzed here. The number of events on each side is counted and shown in Fig. 11. You can see that there are some teams having an almost equal distribution but others with a strong bias into one direction. We currently investigate on the correlations with the system configuration parameters.

To sum up, the history can be a very efficient tool to come to a deeper understanding of people interacting with the Tele-

Board system. Results can be made traceable and implications on team performance become measurable, so that the history infrastructure can be a valuable tool for design researchers as well as designers.
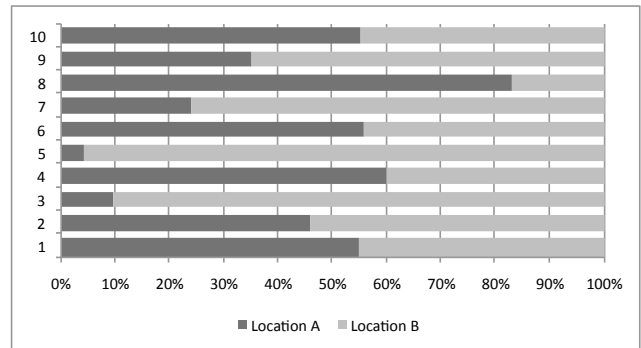

Fig. 11.   Distribution of activity for 10 experiments in a two-location setup

VI. SUMMARY AND OUTLOOK

In this paper we introduced a new approach towards supporting asynchronous communication as well as additional statistics that can be derived from the collected data. On-the-fly communication channel interception leads to fine-grained control on the data capturing process. It enables full reconstruction of every single moment in the communication process. The developed prototype offers the possibility to use digital whiteboard interaction in a synchronous as well as in an asynchronous setting, which enables globally distributed teams to work more efficiently and have a better common understanding in their design processes.

Preliminary research revealed that existing solutions often only focus on one mode of communication, either synchronous or asynchronous (see Fig. 1). The Tele-Board whiteboard client supports both. Features such as resuming work and allowing a hand-over process from one design team to another extents synchronous working modes with asynchronous methods, which turned out to be essential for effective work.

The architecture outline reveals a server-centric approach. It is the most appropriate solution for our use case to keep data at one place. Using XMPP as the communication protocol, this setup can also be integrated into existing message infrastructures. Building on this flexible infrastructure, we developed a variety of tools, allowing interaction with the history data as well as gathering statistical insights on the designers' work.

The Tele-Board history is a new methodology to capture team behavior and their interaction with the used tools. The interaction process can be reproduced and analyzed with the collected data.

For future developments, the hand-over process from one design team to another (possibly in a different time-zone), can be enriched by recording audio or video. The implications of video conferencing for team behavior using our system are currently researched.

We are also elaborating on structural analysis of the history data in order to provide search applications or make meaningful recommendations to the user. The key challenge of this automatic learning from historical data is how to find important points in time. Therefore we have to ask the question: What makes an episode in the history important for the person who wants to understand past design activity? Ongoing user observations and tests will give us more insights on that problem.

The tools described in this paper establish a foundation towards understanding design processes on whiteboards and its consequences on how design activity differs between analog and digital tools. Tele-Board enables designers and design researchers to learn from past design sessions and be more efficient in the end.

REFERENCES

[1] Imagination cubed. http://imaginationcubed.com/.

[2] skrbl: easy to share online whiteboard. http://skrbl.com.

[3] Adobe. Adobe connect. http://www.adobe.com/acrobatconnectpro.

[4] T. Brown. Design thinking. *Harvard Business Review*, June 2008.

[5] K. Carrizosa, O. Eris, A. Milne, and A. Milne. Building the design observatory: a core instrument for design research. *DESIGN 2002*, 2002.

[6] Cisco. "see hello." the future is here with cisco telepresence. http://www.cisco.com/telepresence.

[7] C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1), 1991.

[8] K. Everitt, S. Klemmer, R. Lee, and J. Landay. Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 553–560. ACM New York, NY, USA, 2003.

[9] U. Farooq, J. M. Carroll, and C. H. Ganoe. Supporting creativity in distributed scientific communities. *Conference on Supporting Group Work*, 2005.

[10] W. Geyer, J. Vogel, L.-T. Cheng, and M. Muller. Supporting activity-centric collaboration through peer-to-peer shared objects. *GROUP*, pages 115 – 124, 2003.

[11] Google. Google talk. http://www.google.com/talk/.

[12] Groupworld. Powerful, cost-effective, secure web conferencing software. http://www.groupworld.net/.

[13] R. Gumienny, O. Böckmann, C. Willems, M. Quasthoff, L. Gericke, and C. Meinel. Verteiltes design thinking mit teleboard. *Proceedings "Mensch und Computer 2009"*, 2009.

[14] T. Herrmann. Design Issues for Supporting Collaborative Creativity. In *Proc. of the 8th Int. Conf. on the Design of Cooperative Systems*, pages 179–192, 2008.

[15] O. Hilliges, L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz. Designing for collaborative creative problem solving. *Creativity and Cognition*, 2007.

[16] Inkscape. Inkscape. draw freely. http://www.inkscape.org/.

[17] H. Ishii and M. Kobayashi. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 525–532. ACM New York, NY, USA, 1992.

[18] R. Johansen. *GroupWare: Computer Support for Business Teams Export GroupWare: Computer Support for Business Teams*. The Free Press, 1988.

[19] S. R. Klemmer, M. W. Newman, R. Farrell, M. Bilezikjian, and J. A. Landay. The designers' outpost: a tangible interface for collaborative web site. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 1–10, New York, NY, USA, 2001. ACM.

[20] A. Milne and T. Winograd. The iLoft project: A technologically advanced collaborative design workspace as research instrument. In *Proceedings of the 13th Annual Intenational Conference on Engineering Design (ICED03)*, pages 1–10. Citeseer, 2003.

[21] J. S. F. Network Working Group. Rfc 3921 - extensible messaging and presence protocol (xmpp): Instant messaging and presence, 2004.

[22] Polycom. Polycom telepresence solutions. http://www.polycom.com/telepresence.

[23] Psi. The cross-platform jabber/xmpp client for power users. http://psi-im.org/.

[24] I. Realtime. a real time collaboration community site. http://www.igniterealtime.org/.

[25] T. Rodden. A survey of CSCW systems. *Interacting with Computers*, 3(3):319–353, 1991.

[26] SMARTTech. Smart technologies. http://www.smarttech.com.

[27] A. Tang, J. Lanir, S. Greenberg, and S. Fels. Supporting transitions in work: informing large display application design by understanding whiteboard use. *Conference on Supporting Group Work*, 2009.

[28] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative Coupling over Tabletop Displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 2006)*, pages 1181–1190, 2006.

[29] J. C. Tang and S. Minneman. VideoWhiteboard: video shadows to support remote collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 315–322. ACM New York, NY, USA, 1991.

[30] J. C. Tang and S. L. Minneman. Videodraw: a video interface for collaborative drawing. *ACM Trans. Inf. Syst.*, 9:170–184, 1991.