Understanding asynchronous design work - segmentation of digital whiteboard sessions

Lutz Gericke
Hasso-Plattner-Institut Potsdam
Prof. Dr. Helmert Str. 2-3
14482 Potsdam, Germany
Email: lutz.gericke@hpi.uni-potsdam.de

Christoph Meinel
Hasso-Plattner-Institut Potsdam
Prof. Dr. Helmert Str. 2-3
14482 Potsdam, Germany
Email: meinel@hpi.uni-potsdam.de

***Abstract*— Asynchronous work settings demand for certain hand-over processes - often performed by documentation of the work. This task is challenging especially for creative work, because finding the right form of documentation - including decisions that have been made and explanations how people came to solutions - is non-trivial. Capturing the whole process and replay it afterwards to distributed team members could solve those problems, but is hardly realizable in terms of time consumption. Our approach uses the complete capturing of a session - exemplary implemented using a digital whiteboard system - in order to find out phases of work. This enables us to point out relations between different phases, which can tell us what part was more important than another. We found out that the definition of time slices consisting of certain parameters describing the process can be aggregated into segments. Those segments are classified using an SVM approach, which turns out to give promising results. The overall contribution is an approach, which can be generalized for a variety of captured parameters to allow a precise classification of segments related to the respective overall task.**

***Index Terms*— asynchronous digital whiteboard, segmentation, classification, design processes**

I. Introduction

Communication over distances is still a problem many collaborative work settings are suffering from. Synchronous issues have been researched a lot and industry is developing more and more sophisticated tools to built up an atmosphere of togetherness. There is a large variety of those tools - from human-scale video conferencing to instant messaging clients.

Other issues in companies day-to-day work are not supported that well. Looking at a typical software development team which is distributed between multiple locations - let's say USA, Europe and Asia, simple problems such as time shift issues demand for extensive coordination efforts. Oftentimes, meeting minutes or other kinds of documentation items are created. Although many (video) conferencing systems offer recording functionality, usage of this feature is uncommon. The main problem is that hardly anyone is willing to watch many hours of video in order to understand the past, build up on the findings and decisions of colleagues, and continue with their own work. This is cumbersome, because there is a lack of artifacts pointing out phases that are more important than others.

That is why remote meetings will usually not focus on working together and creating shared content, but more on information exchange, such as updating on the latest developments, new strategies etc. More and more companies (not only limited to IT industry) try out various methods, which aim at bringing a new creative spirit into their companies. One prominent method to do so is design thinking as a toolset for creating empathy for the context (e.g. users of the product or service) of the problem, being able to find creative ideas, as well as rationally analyzing the problem domain. There is a process model visualizing the different stages of design thinking activity. In general, there is a combination of divergent (widening the solution space) and convergent (narrowing down the solution space) thinking. Creative activity often lives from its unstructured nature and is thereby not easy not analyze.

The concept of convergent and divergent activity can be a good starting point for the definition of phases during the creative process. The overall concept of segmentation that is presented in this paper will be exemplary implemented within the Tele-Board research project[1]. This is a platform which supports people during their creative work and is optimized for the use on digital whiteboard devices, but can also be used on regular personal computers.

II. The Tele-Board System - a platform for digital design thinking

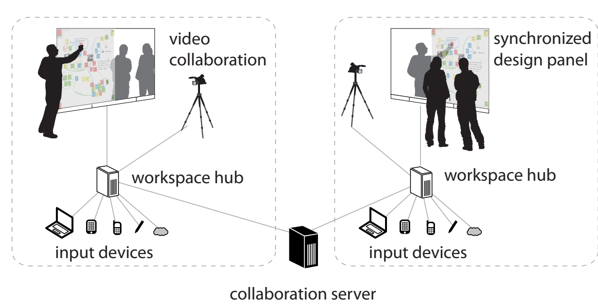

Fig. 1. General setup of the Tele-Board system

Tele-Board is a software system that supports remote collaboration using electronic whiteboards. The interaction with

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

the system is realized in a similar way to conventional whiteboards, i.e. writing, drawing, and erasing on the whiteboard surface can be done in the usual way. Beyond that, it is possible to create digital sticky notes using the whiteboard or additional input devices such as Tablet-PCs, iPads or smartphones. At the whiteboard, users edit sticky notes, move, resize, and generate clusters of the created content.

Remote collaboration is facilitated by connecting several digital whiteboard devices at their corresponding locations with the help of the Tele-Board system as shown in Figure 1. All of the actions are synchronized automatically and propagated to the connected whiteboard clients. Every user can manipulate all sticky notes and drawings, no matter who created them. Furthermore, a videoconference feature is included. The whiteboard content can be displayed transparently on top of the full screen video of other team members. Local team members can see the actions and pointing gestures of the remote team members and vice versa, which facilitates an easier and more interactive session. The flexible architecture of the Tele-Board system makes it possible to start the whiteboard software on every computer. Content created with Tele-Board is organized based on Projects. A project can be used to embrace all phases of a design process. During the course of a traditional design project, a set of analog whiteboards is filled with sticky notes and handwriting. In Tele-Board, the digital counterpart of a physical whiteboard is called Panel.

A. Tele-Board Components

The functionality of the Tele-Board software system is divided among different components, which are as follows:

- *Web Portal:* The web application serves as an administration interface enabling users to maintain their projects and associated panels through a web browser. The whiteboard client that allows editing of a panel is started from this interface, what makes the web application the entry point of the Tele-Board system.
- *Whiteboard Client:* The Tele-Board whiteboard client is a platform-independent Java application. It facilitates whiteboard interaction, e.g. writing with different colors, erasing, and the creation of sticky notes. The client software runs on the users computer, which can be connected to an electronic whiteboard. The whiteboard client interacts with the Tele-Board server component by synchronizing with other clients started at a remote location.
- *Sticky Note Pad:* This component can be used as a dedicated input tool as shown in Figure 1. To increase flexibility in terms of input variety, we created different applications for writing sticky notes as an equivalent to paper-based sticky note pads (running on iOS devices and Android tablets).
- *Server Component:* The server component coordinates all communication between the remote partners. All interactions are transferred as XMPP messages to keep the connected whiteboards synchronized. Storing and resuming capabilities are implemented here using a central relational database.

B. Tele-Board History

As we learned from user feedback and interviews, people in remote teams often work asynchronously. To support these working modes, we developed a solution helping team members, who cannot be connected at the same time, to understand what the others were doing and easily handover their work. Easy navigation through different whiteboard states and resuming work at any previous point in time is realized by the Tele-Board History. A digital whiteboard solution can 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. Another argument for the importance of implicit documentation is the statistical relevance for people researching team behavior and how design over distances and time differences is carried out. The possibilities of the Tele-Board system in terms of the traceability of remote work concerning researchers are shown in [1].

Figure 2 shows an excerpt of the data stored within our system. Every whiteboard event carries the event type, which can be one of the basic operations NEW (element creation), CHANGE (element modification), DELETE (element removal) as well as additional operations such as PICK (cut functionality) or QUEUE (add element to queue of content send by mobile devices). A timestamp annotated to each event keeps the order of each event in its relation to all other events. The XML-encoded description of the whiteboard element itself controls the whiteboard element representation. This data is used to build a segmentation model as described in section 4.

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. 2. Excerpt from whiteboard history table

III. Understanding creative work - What is Design Thinking?

The term Design Thinking is now used for about two decades. The Design Thinking Research Symposium, held first in 1991 by Cross, Roozenburg and Kees, had a significant influence on the formation of that term[2]. In the United States, Design Thinking became know by the work of IDEO, which is a design and innovation consultancy that was established as a fusion of three design firms - David Kelley Design, ID Two and Matrix Product Design[3].

The phrase Design Thinking symbolizes a mindset, an attitude towards certain problems and solving design challenges. Design is meant as the process of coming from a need to an elaborated idea that solves the users problem.

Often Design Thinking is expressed as a process model (see Figure 3) to explain the different steps of understand - observe - point-of-view - ideate - prototype - test. Several iteration cycles are part of the framework, addressing the solution of so-called design challenges (see [2]).

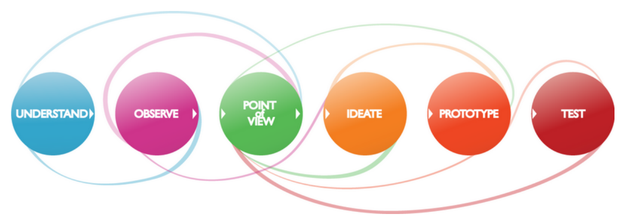

Fig. 3. Design Thinking Process model

Convergent and divergent thinking or activity is a categorization (see [3], [4]), which better describes the way of working within distinct phases. While divergent thinking widens the solution space, convergent thinking usually narrows down the solution space and moves towards extraction of concepts and the commitment to certain ideas.

IV. Segmentation of archived whiteboard sessions

The overall objective is to find substantial changes in working modes, to later be able to point out which part of the process was more important than another. We want to achieve and accuracy during the phase identification of around 30 seconds rather being exact to the second.

As we have shown, our system stores every single operation on the whiteboard. That means every change to a whiteboard element (e.g. a sticky note) will be archived in the database. This granularity is much finer than we want to express it for the phase distinction, but the sheer existence of multiple events of one type can lead to conclusions of what kind of activity has taken place during a time slice. We can configure the length of a time slice. Typical values during our tests were between 30s and 2min and are constant within one analysis run.

A. Datasets

We applied our procedure to two different datasets. Dataset A consists of 10 sessions (panels). Each of those sessions lasts for 40-75 minutes. Overall there are 21674 events in the database for this experiment. The experiment setup was as follows: two participants in each team work remotely (one person at one location) to solve a logic grid puzzle were they had to discuss on the given facts of the puzzle and their conclusions from that. This experiment and early evaluations on team performance are explained in [1].

Dataset B is derived from 19 different panels. These are panels taken from 5 teams with 4 participants each working in a co-located setting. There are 72725 events collected from this experiment. The teams were asked to work on a design thinking challenge for about 5 hours (plus a 1 hour lunch break). They walked through the different stages of the design thinking process, such as user research, information synthesis, ideation, and prototyping.

As you can see from the pure number of events we have a very fine-grained view on the people's activity. Scenario A and B have been very different in their ways of working:

In scenario A the participants worked less creative but on a very specific task. From observations and feedbacks with the participants we found out that they basically worked in three different modes: exploration (reading, understanding, ordering the given facts), fact-centric infilling (straightforward infilling of facts into the grid), and deductive solving (infilling based mainly on conclusions). We decided to use these three working modes as classification labels.

Scenario B shows typical design thinking activity, where it makes more sense to differentiate between divergent working modes, where the solution spaces is broadened, and convergent work, where people synthesize information created in divergent phases in order to come up with concepts, such as a point-of-view or a specific design (see [3]). We decided to use the 2-label classification (divergent/convergent), due to more selectivity compared to classifying the different design thinking phases, which sometimes are hard to distinguish - even for human beings.

B. Segmentation workflow

We define key parameters for every phase, which are directly derived from the number of whiteboard events in the database. Those parameters are defined as a combination of whiteboard element type (sticky note, cluster, path, etc.) and operation code (NEW, CHANGE, DELETE etc.). Counting the number of the equal events within one time slice gives the parameter value. The ratio of those n different parameters forms specific characteristics for each time slice. Therefore we define a time slice s by its parameters p:

$$s_i = (p_0, p_1, ..., p_n)$$

A complete whiteboard session w is the ordered collection of m time slices:

$$w = (s_0, s_1, ..., s_m)$$
$$= ((p_{0,0}, ...p_{0,n}), (p_{1,0}, ...p_{1,n}), ..., (p_{m,0}, ...p_{m,n}))$$

This is the basis for a simplified understanding of the working mode of the people interacting with the whiteboard.

[2]http://design.open.ac.uk/dtrs7/

[3]http://www.fundinguniverse.com/company-histories/IDEO-Inc-Company-History.html

In our application we compare consecutive time slices with each other, when the aim is to find out substantial working mode changes. This approach is more valuable than cross-comparing every time slice. For us it is more meaningful to identify a significant change in working modes at a point in time than identifying two equal working modes at completely different stages in their work.

The general idea is to pairwise analyze different states and group them by their similarity. This similarity measure consists of two components: their similarity depending on the parameters, but also on the temporal difference. If two activities have very similar, but there was a break of some hours in-between, it is a hint that those two time slice should not belong into one phase. So the overall similarity measure is the minimum of the similarity based on temporal aspects and the similarity based on the parameters.

$$sim(s_x, s_y) = min(sim_t(s_x, s_y), sim_p(s_x, s_y))$$

While the temporal similarity is based on a simple linear interpolation (larger difference between end and start of two time slices means lower similarity) and can be configured in its effect, the similarity measure for the parameter similarity is based on the Pearson correlation distance:

$$sim_p = (1 + correlation_{Pearson}(s_x, s_y))/2$$

Based on the similarity values, there is a certain fixed threshold (typically between 0.4 and 0.7), which helps to decide if two time slices belong to one segment. The result of the process is a list of segments, which consist of time slices, each of them defined by a set of parameters. Below, you can see the console output of the segmentation process:

```
Time slices:
...
2010-04-26 13:35:00.0, [2.0, 0.0, 0.0, 0.0]
2010-04-26 13:36:00.0, [1.0, 9.0, 0.0, 0.0]
2010-04-26 13:39:00.0, [0.0, 3.0, 0.0, 0.0]
2010-04-26 13:40:00.0, [31.0, 0.0, 0.0, 0.0]
2010-04-26 13:41:00.0, [2.0, 0.0, 0.0, 0.0]
2010-04-26 13:42:00.0, [38.0, 0.0, 0.0, 12.0]
2010-04-26 13:43:00.0, [18.0, 0.0, 6.0, 0.0]
2010-04-26 13:45:00.0, [0.0, 8.0, 0.0, 0.0]
...

Segments:
start=2010-04-26 13:36:00.0 end=2010-04-26 13:39:00.0 length=240 events=2 class=2
start=2010-04-26 13:40:00.0 end=2010-04-26 13:43:00.0 length=240 events=4 class=0
```

Only a small part of the overall data is listed here. It shows the time slices for a dataset based on only four different parameters (sticky note create, sticky note move, whiteboard move, path draw), which were selected as the most descriptive attributes for this session set. Other test sets use more parameters from the database.

One can see that the correlation-based approach separates between segments of significantly different characteristics. One segment shows a high degree of whiteboard and sticky note move operations and thereby is different from a slice when paths have been drawn. Figure 4 illustrates the complete process.

Fig. 4. Processing steps: segmentation into time slices; merging into phases

V. Evaluating phases

The segments that we got from the first processing step are valuable in terms of which time slices belong together, where have been similar actions taken place. In another step we want to find out, what is the general meaning of these phases. As we want to be as generic as possible, we base the classification on the same parameters, as we use it for the segmentation. Using this approach, we assume that other event streams can be analyzed in a similar manner.

As a classification method, we use a support vector machine (SVM) approach [5]. As we mentioned in section IV-A, there are two different datasets. We used both to train a separate SVM. Time slices and segments have the same parameters, which would allow us for example to train the SVM with the time slice parameters and apply the classification on the segments. To evaluate, if our approach generates meaningful classifications we first did a complete manual classification of all segments.

A. Finding appropriate classes

Because tasks were different in both dataset scenarios, we also defined different classes for both datasets. As we have mentioned in section IV-A, there are specific classes for each scenario:

Scenario A - logic grid puzzle:

- class 0 - exploration
- class 1 - fact-centric infilling
- class 2 - deductive solving

Scenario B - design thinking process:

- class 0 - diverging
- class 1 - converging

Why is it necessary to define different phase labels for different scenarios? The way people are working with a collaborative system such as the Tele-Board is as different as the content as they are working on. We can abstract in a way that for example in the logic grid experiment, it was not necessary to create any sticky notes, so that we can better adapt to the given task and thereby focus on drawing and whiteboard and sticky move actions. Scenario B was much wider in its scope, which lead us to consider each of the 14 archived parameters.

B. Training the SVM

Our implementation is based on Java using the libSVM Implementation of Java-ML. From our dataset A we found 108 different segments. This equals an average segment length of 5.76 minutes. Those segments where classified manually to later train the SVM with that data. In this dataset we used 4 different parameters.

From dataset B we found 275 segments, which represents an average segment length of 5.94 minutes. We adjusted the parameters to almost match the average segment length in order to enable better comparability. This dataset consisted of 14 different parameters.

In order to evaluate the expressiveness of the classification done by the trained SVM, we do cross-validation on each dataset.

C. Cross-validation of phase classification

The overall goal is to find a fitting classifier for certain situations, such as scenario A and B. Currently, we test it with those two datasets, which are different in terms of what is going to be classified. On the one hand we want to assess convergent and divergent behavior, on the other hand it is about the way of solving a logic grid puzzle. This does not allow us to use the data from one experiment to train the classifier and apply it on the other dataset. That is why we used a cross-validation approach to validity of our data towards classification using a support vector machine.

We applied a 10-fold cross validation on both scenarios. It turns out that the error rates achieved using SVM tuning[4] are quite promising. In scenario A we achieve an error rate of 24.3%, whereas scenario B gives us an error rate of only 11.8%. It is important to mention that in scenario A the dataset consists of about 2.5 times less instances (108 vs. 275) and - potentially even more important - of three classes, whereas in scenario B there are only two different classes. Additionally, there are only four different parameters for A - even though those are the most characteristic ones - and 14 parameters for B. When merging classes 1 and 2 for scenario B (fact-centric infilling and deductive solving combined as "solving"), we get a error rate of about 13.7%.

VI. Interfaces for Phase-Based Navigation

The results of the cross-validation using the SVM as a classifier encourages us to further elaborate on our approach. Using the classified segments, a bunch of different options are imaginable. We could enrich the datasets with further parameters, which could give us even more conclusions on the segment's relevancy. This additional data could come from different sources. First, we could use automatically gathered

[4]http://cran.r-project.org/web/packages/e1071/e1071.pdf

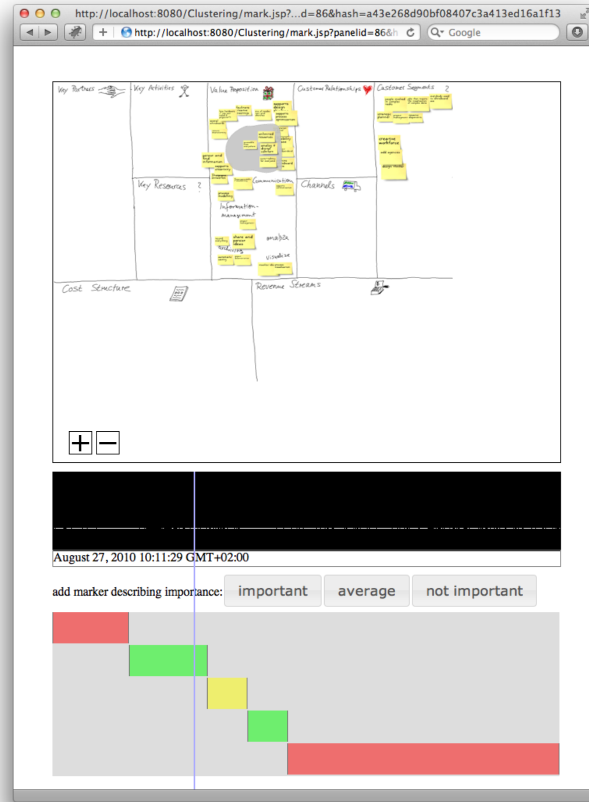

Fig. 5. Phase Evaluation - whiteboard session can be scrolled through and markers are added to show the overall importance of phases

data, which could come from different sources: video analysis of the capture video conferencing, sensor data attached to the participants or feedback from people exploring the history of a whiteboard session.

Last point is the most promising factor, because our idea is to make use of the feedback for each of the recorded sessions. That means, we gather feedback on the validity of our segmentation and classification and use that feedback on future session segmentation and classification. A first assumption is that very short segments are almost irrelevant and can be left out in the next step. This can be either direct feedback, when the user actively evaluates the segment or it can be indirect feedback, were we evaluate the segments value based on the number and duration of people looking at it. Other factors can also be taken into account. A possible interface is shown in figure 5, whereas an interface to the exploration view is shown in figure 6.

VII. Outlook

Our approach creates the foundation for a tool allowing remote teams to faster navigate through the history of colleagues' work. It is easily adaptable to a wide range of input data. Different sources could be combined into the time slices and thereby more information will be taken into account when

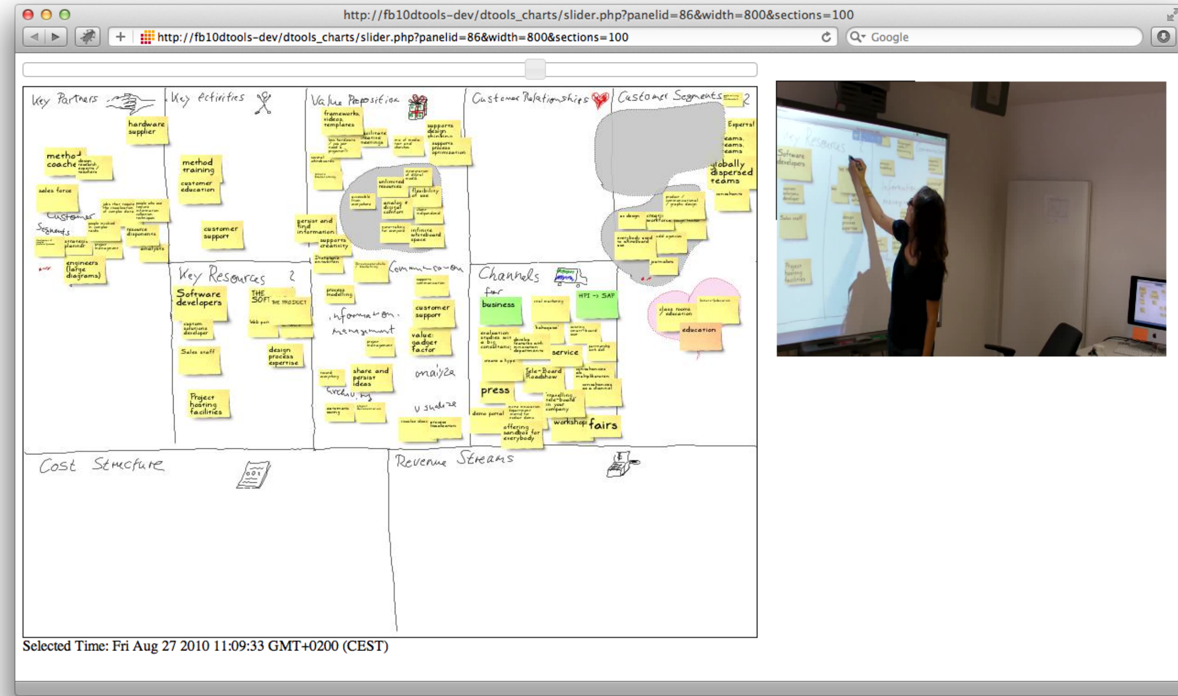

Fig. 6. Combined and synchronized video and whiteboard content view (read-only view)

basically saying, this phase is important or not. The most obvious idea could be to integrate sensor data into the system. Other publications ([6], [7], [8]) have looked at different sensors in order to infer the kind of activity a person is taking out and found accelerometer data as the most significant source of information. Beyond that, accelerometers are omnipresent in smart phones, MP3 players etc. We are planing on further experiments, which will combine this data with the whiteboard archive.

There are elaborate approaches in research and practice allowing us to automatically "understand" what has happened in the video. One can think of basic analyses such as determine an activity level at each scene or become more complex and look at which people where involved and what kind of motions did they execute.

To enrich the segmentation and classification data, we think of two different kinds of feedback: direct and indirect feedback. Direct feedback would ask the use to rate what he has seen, e.g. a phase that was suggested to be important could be rated as helpful or not. Indirect feedback would look at how people browse the history of a panel. Points on the timeline when many people looked at are potentially more important than others which many users ignored. A combination of both kinds of feedback suggests to be a valuable model for enriching analysis results.

Summing up, we found a way of defining a structure within unstructured design processes and propose a way of analyzing those processes. It is hard to say, if we ever completely comprehend what happened during one of these processes, but we are giving assistance to easier navigate through historic work.

Acknowledgment

The authors would like to thank the support of the HPI-Stanford Design Thinking Research Program.

References

[1] L. Gericke, R. Gumienny, and C. Meinel, "Analyzing distributed whiteboard interactions," in *CollaborateCom 2011*, 2011.
[2] T. Brown, "Design Thinking," *Harvard Business Review*, Jun. 2008.
[3] T. Lindberg, R. Gumienny, B. Jobst, and C. Meinel, "Is there a need for a Design thinking Process?" in *Design Thinking Research Symposium DTRS8*, 2010, pp. 243–254.
[4] D. S. Kerr and U. S. Murthy, "Divergent and Convergent Idea Generation in Teams: A Comparison of Computer-Mediated and Face-to-Face Communication," *Group Decision and Negotiation*, vol. 13, no. 4, pp. 381–399, 2004.
[5] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition*, 1974.
[6] P. H. Veltink, H. B. Bussmann, W. de Vries, W. L. Martens, and R. C. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers." *IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 4, no. 4, pp. 375–85, Dec. 1996.
[7] A. R. Doherty, A. F. Smeaton, K. Lee, and D. P. W. Ellis, "Multimodal Segmentation of Lifelog Data," no. 2006, 2007.
[8] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions," *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pp. 113–116.