Behavior Assessment and Visualization Tool

Deepak Jagdish
School of Literature, Communication and Culture,
Georgia Institute of Technology,
686 Cherry St.,
Atlanta, GA 30332 USA
d.j@gatech.edu

Abbas Attarwala
College of Computing,
Georgia Institute of Technology,
801 Atlantic Drive,
Atlanta, GA 30332 USA
aattarwala3@gatech.edu

Ute Fischer
School of Literature, Communication and Culture,
Georgia Institute of Technology,
686 Cherry St.,
Atlanta, GA 30332 USA
ute.fischer@gatech.edu

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Abstract
This paper introduces our work on a new Tablet PC-based tool that allows near-real-time coding (a technique of classification) of video-recorded or live behavior. The tool also allows the user to create and manipulate simple interactive visualizations of the coding results. This tool has been designed both to advance behavioral research and to support applied uses, for instance in professional coaching. We envision that this tool will be extremely versatile as users will be able to classify in near-real-time individual and team-behavior occurring in many research domains including HCI. This paper describes the salient design and interaction aspects of this tool, and the improvements it has over existing systems.

Keywords
Near-real-time Coding, Interactive Timeline, Interactive Visualization, Tablet-based UI, Pie-Menu

ACM Classification Keywords
H.5.2 User Interfaces – *Interaction styles, Input devices and strategies, Graphical User Interfaces (GUI)*

General Terms
Design

Introduction

An important aspect of human factors and HCI-related research is the observation of human behavior in field settings or simulated task environments. Typical research questions are, to name a few, usability testing, characterizing individual and team performance, or assessing and providing feedback to trainees. One of the major challenges researchers face is the fundamental question of how to record observational data. Approaches that involve note-taking provide researchers with a set of rich information; however, coordinating observation and note-taking is unwieldy and taxing to working memory, rendering data collection susceptible to errors like omission of data or an incorrect classification. In recent years, software programs have been developed that allow users to tag their coding (i.e., behavioral classifications) to video-recordings of relevant behavior. While these systems provide considerable advantages over note-taking (such as increased reliability; time-stamped observations linked to behavior), they, too, come with limitations, most notably concerning their usability and flexibility.

The objective of our project is to create an observational tool that, building on the strength of existing technology, overcomes some of their limitations. To this extent, we developed a tablet-based tool that enables users to perform near real-time coding of behavior either from video-recordings or by observing live performance. The tool was designed both to advance behavioral research and to support applied uses, for instance in professional coaching. Several design goals motivated our work: (1) The tool should be easy to use, with the interface facilitating the coding process rather than adding to raters' workload; (2) It should be flexible and not limited to specific research questions, allowing users to define coding categories; (3) It should support the simultaneous assessment of multiple agents acting in concert or concurrently; and (4) It should provide users with an interactive timeline-based visualization of observational data.

Related Work

A lot of the work done until now with regard to coding comes from video annotation tools. A workshop report by Rohlfing et al in 2006 [1] compared the multimodal annotation tools available then. Some of the prominent tools covered in this report were ELAN [2] and ANVIL. ANVIL (Annotation of Video and Spoken Language) is a video-annotating tool designed to support research in Linguistics, Human-Computer Interaction, Gesture Research or Film Studies [3]. It allows the user to annotate a video by combining a basic text entry system with a media player. It allows the user to have multiple tracks on the same video, thereby allowing for observation and classification of different phenomena. It also supports user-defined coding categories. ELAN is a more limited version of an annotation system since it aims to achieve the specific goal of creation of text annotations for audio and video files. One of the major drawbacks with both ANVIL and ELAN is the time taken to perform coding. Insertion of each annotation requires the user to pause the video, then type in the annotation, and this process has to be repeated for every annotation. Also, these tools only aspired to perform annotation of digital audio-video data.
So they did not allow the user to – (a) 'code' actions or events and then view the results in an interactive manner, (b) have structured coding techniques, or (c) support live coding.

FIT-System (flexible interface technique) is a commercial computer-based method for the classification of live events [4]. Using mobile devices as a platform, users define their own workspace interface (including coding symbols) by drawing or writing on a transparency overlay. Coding is done by tapping the appropriate symbol on the display with a stylus. In terms of features offered, a commercial tool called Observer XT [5] is the closest available option to our tool. It comes with a data visualization module, has a mobile version and separate third-party plugins are available for statistical analysis. User-defined coding categories are permitted and even though the codes are mapped to keyboard shortcuts allowing for coding to be done in near-real-time, in the case of multiple coding passes, it might be difficult for users to remember key-code pairings.

Apart from the tool-specific drawbacks mentioned about each existing system, they all share other drawbacks such as: (1) None of them seem to support cross-classification of behavior to agent; (2) Most of these tools seem to be designed to deal with specific research questions; and (3) They all have tedious and cluttered interfaces which interfere with the user's experience of coding. We aim to resolve the drawbacks found in these existing systems, build on their strengths and introduce new features to make a full-fledged coding and visualization tool.

System Design of the Tool

The current prototype of the tool was designed to support the analysis of video-recorded team interactions that occurred during computer-simulated search missions set in Antarctica. Teams (shown in screenshots in subsequent sections) consisted of four distributed members identified by different colors (red, blue, green and purple). Analyses focus on the cognitive and social aspects of team communication such as task coordination and interpersonal affect [6].

Salient Design & Interaction Features

Fluid User-Tool Interactions

This tool has been specifically designed for use on a Tablet PC exploiting the new facilities that touch-based interfaces provide. However, it will also run on laptops and desktop computers so that the user has the freedom to rely on stylus, mouse or fingers as the input device. This allows for a much more fluid interaction between the user and the system, and supports our novel coding technique of tapping a category in a pie-menu (see description below in the section titled *Coding by Pointing*).

Coding as a Pointing Action

The unique feature of our coding tool is that each agent is "assigned" an individual space on the input screen (see Figure 1). This design feature is consistent with human language processes –and thus should be intuitive to users—as it is derived from sign languages. Signers place the sign referring to a person or object in space and convey pronominal reference to this person or object by pointing to the appropriate location [7]. Analogously in our tool, coding categories will be overlaid as pie-menus on each space. Consequently, when a code is selected, it will be linked with the "agent occupying the space." This space-based coding style affords users to link ratings to observed agents by performing one fluid movement towards a single location, rather than having to do multiple touches/clicks in different locations of the screen. As

shown in Figure 1 (zoomed version of the upper-left quadrant of Figure 2), tapping the code "Compliment" in the space of team member Blue will be recorded as *"Speaker: Blue; Interpersonal Affect Type: Compliment"*, where the user has configured the coding categories to be types of Interpersonal Affect.

Figure 1. Discrete Coding using a pie-menu

Our pointing-based coding technique allows for simultaneous classification of actions of up to six agents being observed. This number reflects screen real-estate considerations and is within the processing capacity of human working memory [8].

Coding of Live Interactions or from Video-Recordings
Pointing-based coding supports near-real-time assessment of both live as well as video-recorded behavior. We say *'near'-real-time* because there will always be a time-lag between the occurrence of behavior and users' data entry, dependent on users' abilities and the complexity of their classification system. Figure 2 illustrates the input screen used to code interactions of a distributed team. Recordings of individual team members play in the background of their designated spaces outlined by the players' corresponding color. For co-located teams –i.e. when there is one video recording—the recording may be shown on a separate computer screen, or it may be displayed in the center of the tablet screen with the team member "spaces" surrounding it (the location of these spaces can be customized as per the user's needs). In the case of observing live interactions of co-located teams, team members' spaces on screen will be identified only by the outlined color, and the user can draw these spaces such that it reflects the actual spatial configuration of the agents.

Flexible Coding Categories
Users can custom-define the coding categories that appear in the pie-menu. This feature enhances the usability of our tool in many domains and for different research questions. In the tool's current version the number of coding categories is limited to six. Having more categories (i.e., finer distinctions) would likely exceed the processing limits of working memory [8]; in particular, since our emphasis is on near-real-time classification. More coding categories may also impede readability and usability of the interface.

Our tool also allows users to focus on the duration of behavior (= *interval* coding), or to characterize it as discrete instances that occur over time (= *discrete* coding). Prior to coding, users can specify the nature (interval vs. discrete) of the coding categories. Figure 1 shows the interface when users engage in discrete coding. For interval coding, the pie-menu will have a circular "Start/End" button in the center (not shown in

Figure 1) in addition to the different coding categories. To initiate interval coding, users need to tap *Start* (which will then change to display *End*) and then tap on the coding category. To indicate the end of the interval, users need to tap *End*.

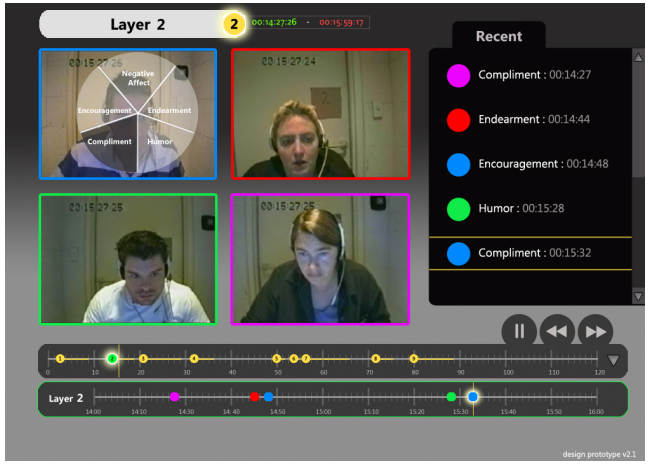

Figure 2. Interface layout of the Coding module

Intuitive Workspace Elements
Several interface elements further enhance the usability of our tool:

(a) *Layered Timeline* – Figure 2 shows a timeline in the lower region of the interface. Its colored oval shapes correspond to observations a coder noted for each of the four team members. To represent multiple iterations of coding of the same video recording, layers were introduced on the timeline; thus each layer concerns a different aspect of team members' behavior. Layers may be independent of each other, or may be in a parent-child relationship (i.e., a later coding pass builds on a previous one). When an observation in the parent layer gets refined by coding in the child layer, its node on the timeline is highlighted (brightened) to represent visually that the current coding is related to it. This display solution provides users with a quick overview of the coding process and enables them to re-visit coded segments since codes are time-stamped and tagged to the video.

(b) *Coding History Panel* – On the right side of the interface shown in Figure 2 is the Coding History Panel which provides users with an up-to-date account of recently created codes. Its simple grid-based design and the fact that agents are identified by different colors, allows users to gain a quick overview of their coding and to verify its accuracy.

Visualization Module
This module allows users to visualize their observations by providing some quick overview options to 'eye-ball' the data. It also allows users to export the data into other data analysis tools, such as SPSS.

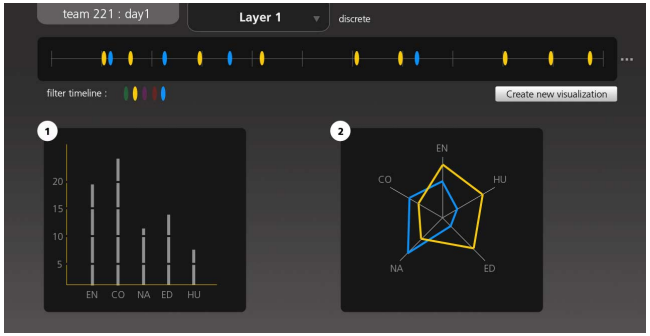

Figure 3. Two visualization widgets created by the user appear side-by-side in the Visualization module of the tool

As shown in Figure 3, the layered timeline appears on top, and the user can select agents that should appear in the Visualization results by just tapping on their respective color in the filter below the timeline. The user can then invoke a menu to create different visualization widgets which includes options such as *Histogram, Pie-Chart, Radar-Map,* and so on. Some existing visualization systems like Tableau [9] use a similar feature which helps the user narrow down the options of visualization depending on the data-set being used.

Future Development and Conclusions

The current version of the tool does not use any automated functions. However, future versions could use intelligent automation for functions like Segmentation of video based on presence of physical activity/voice, Latent Semantic Analysis [6] and tracking of moving entities in a video feed. Our next step will be to test our current prototype with researchers working in different domains that are rather different from team communication analysis, for instance an improv-theater group that wishes to analyze their acting sequences, or with researchers evaluating interaction patterns of a new software prototype, to see whether this tool's interface can stand the test of versatility. We are also exploring the use of symbols, apart from colors, to distinguish between agents that may be better suited for other research domains.

Some preliminary interface-related feedback indicated changes to be made to the visual characteristics of some elements like the width of the timeline, coded events appearing on it, better visibility for outlines of agents, transparency of pie-menus depending on their usage history etc. These changes are being incorporated in the next version of the tool.

Acknowledgements

This work was supported by NASA Cooperative Agreement NNX08AW90A to Ute Fischer.

References

[1] Rohlfing, K. et al, "Comparison of multimodal annotation tools – a workshop report", *Gesprächsforschung Online-Zeitschrift zur verbalen Interaktion,* 2006, pp 99-123

[2] Hellwig, B., van Uytvanck, D., & Hulsbosch, M. 2009. *ELAN – Linguistic Annotator, version 3.7*. Available at: http://www.lat-mpi.eu/tools/elan/

[3] Kipp, M. Anvil – "A generic annotation tool for multimodal dialogue", *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, 2001, pp 1367-1370.

[4] Held, J., & Manser, T. "A PDA-based system for online recording and analysis of concurrent events in complex behavioral processes," *Behavior Research Methods*, 37 (1), 2005, pp 155-164.

[5] Observer XT available at : http://www.noldus.com/

[6] Fischer, U. "Enhancing team performance in exploration missions", *Second Year Report on NASA Cooperative Agreement NNA05CS50A* : Georgia Institute of Technology, 2007

[7] Emmorey, K. & Falgier, B. "Conceptual locations and pronominal reference in American Sign Language", *Journal of Psycholinguistic Research*, *33(4)*, 2004. pp 321-331.

[8] Miller, G. A. "The magical number seven, plus or minus two: Some limits of our capacity to process information". *Psychological Review,* 1956, pp 81-97.

[9] Tableau Software available at : http://www.tableausoftware.com/