ABSTRACT
To be most useful, evaluation metrics should be based on detailed observation and effective analysis of a full spectrum of system use. Because observation is costly, ideally we want a system to provide in-depth data collection with allied analyses of the key user interface elements. We have developed a visualization and analysis platform [7] that automatically records user actions and states at a high semantic level [8-10], and can be directly restored to any state. Audio and text annotations are collected and indexed to states, allowing users to comment on their current situation as they work, and/or as they review the session. These capabilities can be applied to usability evaluation of the system, describing problems they encountered, or to suggest improvements to the environment. Additionally, computed metrics are provided at each state [8, 10, 21, 22]. We believe that the metrics and the associated history data will allow us to deduce patterns of data exploration, to compare users, to evaluate tools, and to understand in a more automated approach the usability of the visualization system as a whole.

Categories and Subject Descriptors
I.3.6 [Computer Graphics]: Methodology and Techniques – interaction techniques, metrics, usability, standards.
H.5.2 [Information Systems]: User Interfaces – theory and methods, standardization, user interface management systems (UIMS), evaluation/methodology, graphical user interfaces.

General Terms
Measurement, Documentation, Experimentation, Human Factors, Standardization, Verification

Keywords
User monitoring, session history visualization, session history analysis.

1. INTRODUCTION
To be most useful, evaluation requires metrics to compare and classify operations between systems, users, tasks, and procedures. These metrics must be based on detailed observation and effective analysis of a full spectrum of system use. Because manual observation and analysis are extremely costly, ideally a system would provide in-depth data collection with allied analyses of the key user interface elements. The more specialized an environment is, the more tailored the observations and analyses can be. We focus on visual analytics and specifically the exploration process.

There have been many implementations of recording for usability evaluation of GUI applications [6, 11, 12, 14-16, 20, 25, 26]. There have also been a small number of recording and/or annotation environments targeted towards visualization in particular, such as InfoVis VDM [19], the work of Jankun-Kelly and Ma [17, 18, 24], and Loughlin et al’s annotations of scientific visualizations [23].

Our research group has developed a visualization and analysis platform [7] that supports a variety of visualization and analysis tools for exploring high-dimensional data (dimensions well over a thousand). There are very few systems providing interactive visual high-dimensional exploration, and as such comparative studies into usability are difficult.

However this system also automatically records all user actions and the resulting system states at a high semantic level. Therefore it can be used to compute metrics and provide automatic support for evaluating these tools [8-10], incidentally using the system itself to visualize and analyze the history data and metrics.

Several components of the system support these functions:

1. The system records its state throughout the session, and can be directly restored to any past state,
2. The recorded history can be visualized as data,
3. Audio and text annotations are collected and indexed to states, and
4. Process metrics are computed at every state.

The combination of viewing history, restoring states, and viewing and searching audio and speech-to-text annotations provides for very flexible annotation and review. Users can comment as they work, or while visualizing their session and/or replaying selected portions and exploring – even taking new actions they didn’t think of before.
Computed process metrics which are provided at each state [8, 10, 21, 22] and the associated history data allow us to deduce patterns of data exploration, to compare users, to evaluate tools, and to understand in a more automated approach the usability of the visualization system as a whole.

These components allow us to provide detailed observations with minimal impact on user tasks – ranging from automatic monitoring only, to spoken annotations as they work, to deliberate bookmarking and adding comments about important points in the session, to restoring previous states and possibly annotating further. We believe this range of recording and documenting choices will lead to better understanding of the exploration process we are trying to analyze and model. We believe that the metrics and the associated history data will allow us to deduce patterns of data exploration, to compare users, and to evaluate tools. By correlating results with metric performance standards, we can perform principled analysis of a variety of different exploration tools and fine tune these. Metrics also allow an automated approach to determining many aspects of the usability of the visualization system.

2. VISUAL HISTORY METRICS
Our research group’s architecture for session recording and analysis has been implemented in a data visualization and analysis platform used both by data analysts and data exploration researchers. It supports the recording of in-depth session histories, and the application of those histories in many ways to benefit users and facilitate analysis. Tools automatically record and interact with session histories. The session history infrastructure provides automatic recording, restoring, annotation, computed process metrics, visualization and analysis of user actions. These capabilities provide a basis for some novel system evaluation techniques.

Our system’s primary theoretical basis is an extension of the Generalized Data Exploration (GDE) model of J. P. Lee [21]. User actions are recorded as changes to system state. Each action is categorized in a data exploration taxonomy, so state changes anywhere in the system can be compared in a principled way [9]. For example, a user or researcher might choose to ignore actions that only changed data presentation (not shown), or to only look at the state changes of a single tool (for example, see Figure 3).

The GDE Model is based on determining unique “data entities” (system states) and “derivations” (transitions) between them. Metadata including audio and text annotations may be associated with the states in which it was recorded. Users or researchers reviewing a session history may also add annotations to a previous state, or restore it and continue exploring (i.e. adding branches to the history tree) from there.

A description of what users and researchers can do with session history as a co-explored data set was previously described in detail [8, 10]. Here we describe the computed process metrics and their possible use in more detail.

Because audio and text annotations are correlated with actions, evaluation protocols like “thinking aloud” can be analyzed in the context of the user actions and system states that accompanied the words. Couple this with restoration and now users can not only comment on their current situation for future reference (the typical use of annotation) but also review their sessions through multiple passes in order to provide additional comments as they go over the session. Typically these features are used to document explorations for coworkers or future reference, but they can also be applied for usability evaluation of the system, for explaining problems having been encountered or for suggesting other useful and supportive items within the environment.

Users or researchers viewing a session may alternate freely between browsing and searching annotation text, viewing patterns in the history visualized as data, and restoring and navigating in the history. These capabilities provide a superset of the interactions required for concurrent or retrospective verbal protocols and a variety of related usability evaluation techniques [4, 5, 13, 28].

3. INTERACTING WITH HISTORY
The system allows interactions with history by monitoring, restoring, and replaying previous sessions. Audio/text annotation is also linked to history bidirectionally (view or restore history state→read/listen to concurrent annotations; or search annotations→restore the history state they describe). A brief summary of monitoring, restoring, and annotation follows; for more details see [10].

3.1 Restoring Previous States
We use Rekimoto’s term “Time Travel” for return to previous states [27], often referred to by the more limiting terms “undo and redo” [1-3]. We restore states rather than repeat or undo actions. This provides greater flexibility as users can time travel directly to any point in the session. For example, when viewing audio speech-to-text and typed annotations, it is easy to put the system into the state in which each annotation was spoken or entered.

Users can visualize history as general data set [8, 10] and use a variety of mechanisms to locate recalled or other interesting points in the history using any of the visualization tools or indexing by text search of annotations. At any time, they can view associated annotations or time-travel the system to that state.

3.2 Voice Annotations
Voice annotation is a natural and unobtrusive means of gathering information from users. This information complements the automatically recorded actions. Users can express their thoughts and findings during the discovery process and the ideas they express form contextual links between their abstract thoughts and the current concrete system state. We use automatic speech to text translation. Annotation runs continuously throughout the session enabling them to annotate any part of the session without a special effort. In the next section we show a prototype of the voice annotation window with (manually-transcribed) speech-to-text annotations, along with a screen shot of the session in which they were spoken.

Annotations are provided as both audio and text. Annotation text can thus be searched for a keyword, and the concrete states that related to those ideas or triggered the utterances can be observed. Obviously user comments about the performance of the software, such as emotional responses to bugs and missing features, are directly useful. Calculating metrics (Section 5) and implementing formal evaluation techniques are more systematic uses of history.
“Thinking aloud” protocols can be contemporaneous or retrospective, with users either speaking as they work or offering explanations as they view a videotape of their actions [5, 13]. Our infrastructure supports either mode, with additional possibilities. Users can interrupt action replay to try other possibilities and browse back and forth as they add to their previous remarks. Users can also visualize the history as data to explain it from an overview perspective. Combining traditional evaluation with information visualization and analysis techniques may find new patterns that are not part of the conventional repertoire.

4. VISUALIZING HISTORY DATA
The history of each user session generates an abstract data structure that we routinely represent as a Session Graph. Figure 1 shows a collection of data visualizations with such a session graph at its bottom, followed by a (simulated) annotation window showing two analysts actual comments during this session.
The Session Graph implements the GDE Forward Derivation Component Graph (FDCG) that only shows the first path into each unique state. Derivations that return to prior states – taking actions and then reversing them – are considered in calculating metrics (Section 5), and may be examined by probing the Session Graph, but are not routinely drawn. The branching history tree structure is represented from left (the start of the session) to right (the “deepest” operation, created by the longest string of actions that explored new states rather than returning to old ones.) Probing a Session Graph vertex reveals all its entries, exits and metrics.

The default Session Graph or other visualization is of the state of the entire system. However, it is often useful to display the states of just part of the system: a tool, a selection of the operation taxonomy, or even an individual control. Figure 3 shows the states of one tool within a multi-tool session. Users choose the data labeling each state and edge.

5. METRICS

5.1 GDE Metrics

The GDE defines two kinds of metrics on the session graph: vertex metrics (Figure 4) and path metrics (Figure 4).

Path metrics, computed upon all or a connected subpath of derivations, include path depth (number of transitions from the source to the node in question) and breadth (number of different states visited in parallel at the same depth from the source).

Figure 4 shows these metrics as well as Forward Progress Rate, the number of forward derivations per minute. Forward progress (vs. raw derivations/minute) is justified in GDE terms by Lee’s [4] Data Refinement Semantic Postulate that the greater the number of steps from the start along a forward path of a vertex, the more refined and closer to a unit of knowledge it is.

5.2 Metric Calculation

The metrics for each model may be viewed as a separate dataset of unique states, similar to the view of its history as a dataset. Like the raw history data, these datasets may be analyzed or visualized by any tool in the platform. This general approach provides interesting flexibility for evaluation (see next section.)

5.3 Metrics for Evaluation

Several of the GDE metrics are directly usable for evaluation. For example, the depth and breadth metrics, and time-based metrics such as exploration rate, indicate whether the analyst is able to make rapid progress with this data and tool set.

The ability to view and compute metrics on the state history of a single component of the system should be especially useful here. If one tool or widget shows much broader paths (indicative of...
users trying many different approaches that probably didn’t work) or slow exploration rates, this may indicate usability problems with that part of the program. The metric dataset for all or selected components can be analyzed visually or statistically to find targets for closer scrutiny. Evaluators may view the state history for questionable components and “time travel” to states where users appeared to have trouble, looking at their annotations, seeing the state of the interface they saw, and stepping forward and back to find where they went wrong. Figures 5 and 6 show how the metrics can be represented on the Forward Derivation Component Graph representations.

![Figure 6: The GDE Forward Derivation Component Graph](image)

Annotations are represented by pluses (+) visible at nodes. Number of pluses represents importance. Mousing over the nodes pops up annotation text. The numbers displayed indicate the step at which the state was first entered. Vertex color is in-degree (blue-cyan). The start vertex has the lowest in-degree, so it is black. Vertex 10 has the highest degree so it is white. There is an apparent cycle in the graph because the user at state 10 went back to state 7 through a change of parameter settings. This happened again at state 12.

![Figure 7: Heatmap representation of vertex metrics](image)

In Figure 6, the left side of the heatmap represents incoming edges and the right the outgoing. Top left is the in-degree of FDCG and the top right the FDCG out-degree. Lower left is total incident and lower right is total exiting edges. Note that incident and exiting count back edges and self-loops which the FDCG does not. Color is red for lowest and green for highest. All counts are normalized individually by maximum for that single item (in and out degree or total incident and exiting edges. Thus vertex 10 top left and right are green because they are the largest for their category. The in- and out-edges below them are actually larger numbers, but they are yellowish because they are not the highest for their category.

6. Conclusions

We described a data exploration environment that provides for monitoring, analyzing and interacting with session history. We described the support for viewing and calculating metrics on the history of the whole system or individual parts, and for voice annotation and search, and its use for evaluation. We gave several brief examples of visualizations of the computed process metrics. This of course is preliminary work.

7. Future Work

We are exploring the use of these metrics and with modeling and statistical analysis of multiple user sessions to develop higher level tool and system evaluations. As we progress to having groups of users share their session histories, these metrics will become more tied to multiple data set explorations (the user’s data being explored and others’ session histories). This will require extending the visualization/exploration taxonomy to include session histories and the development of specialized metrics.

8. ACKNOWLEDGMENTS

The authors thank J.P. Lee for his helpful discussions, and all the members of the UVP team who contributed ideas and implementations to the session history implementation, especially Alex Gee, Chris Lawrence, and Jon Victorine.

9. REFERENCES


[9] Goodell, H., Chiang, C.-H. and Grinstein, G.G. Taxonomized Automated Monitoring of Interactive Visualization, *Computer Science Department, University of Massachusetts at Lowell, Lowell, Massachusetts, 2006*. All rights reserved.