Metrics for Analyzing Rich Session Histories

Howard Goodell, Chih-Hung Chiang, Curran Kelleher, Alex Baumann, Georges Grinstein*
University of Massachusetts at Lowell
One University Avenue
Lowell, MA 01854 USA
*1-978-934-3627

hgoodell, cchiang, ckellehe, abaumann, grinstein@cs.uml.edu

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 [1] that automatically records user actions and states at a high semantic level [2 and 3], 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 [3, 4, and 5]. 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.

We have developed a visualization and analysis platform [1] that supports the exploration process for 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. Since our system automatically records all user actions and states, at a high semantic level, it can be used to compute metrics and support such evaluation [2 and 3].

Several components of the system help in this support:

1. The system can be restored to any viewed state and stepped forward and backward; and
2. History can be visualized as data, and
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 [3, 4 and 5] 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 three components allow us to provide broader and more accurate observations leading 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. This will help us understand in a more automated approach the usability of the visualization system as a whole.

2. VISUAL HISTORY METRICS

Our research group has developed an architecture for session recording and analysis implemented in a data visualization and analysis platform. This platform is used both by data exploration researchers and data analysts. 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 [4 and 5]. 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 [2, 3]. Metadata including audio and text annotations is associated with the states in which it was recorded. Users or researchers reviewing a session history may also add annotations at any state.

A description of what users and researchers can do with session history as a co-explored data set was previously described in detail [3]. 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 [6-8].

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 [3].

3.1 Restoring Previous States

We use Rekimoto's term "Time Travel" for return to previous states, often referred to as "undo and redo" [9]. We restore states rather than deal with repeating or undoing actions. This provides greater flexibility as users can time travel directly to any point in the session. This is necessary for evaluation annotation.

Users can visualize history as general data set [3] 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. Then, they can 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.

Annotations are provided as both audio and text. Annotation text can thus be searched for a keyword, and the state(s) in which the word was said can be retrieved. This procedure allows searching for abstract ideas or remembered terms, 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 users' emotional responses to bugs and missing features, are directly useful to improve system performance and track bugs. However calculating metrics (Section 5) and using this tool to implement more formal system evaluation techniques allows a more systematic use 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 [8]. 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 (see next section.) Combining traditional evaluation with the application of information visualization and analysis techniques may find new patterns that are not part of the conventional repertoire.

4. VISUALIZING HISTORY DATA

Each user session generates an abstract data structure that we can think of as a Session Graph. Figure 1 shows a collection of data visualizations with such a session graph at its bottom.

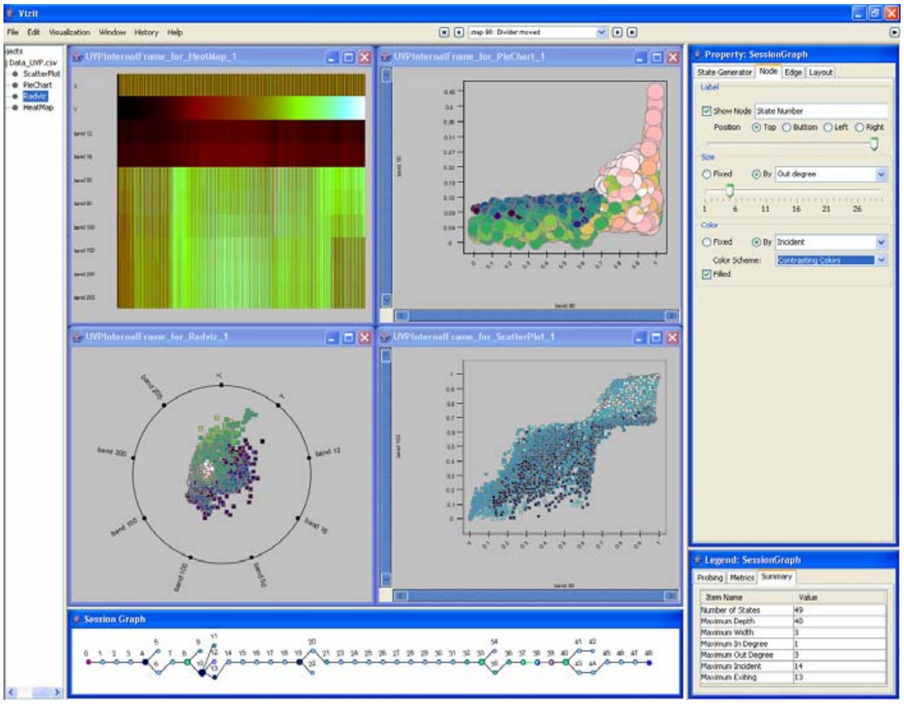


Figure 1: Visualization Session with Session Graph Representation at the bottom. The others (above) are data visualizations,

The Session Graph implements the GDE Forward Derivation Component Graph (FDCG) that only shows the *first path* into each state: returning to prior states – taking actions and then reversing them – is applied to metrics (Section 5), but not drawn. It draws a branching tree structure 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 2 shows the states of one tool, PieChart, within the above multi-tool session. This is a simple graph with a vertex for each state and an edge for each action. Users choose the data labeling each state. More data is available by probing individual vertices with the mouse.

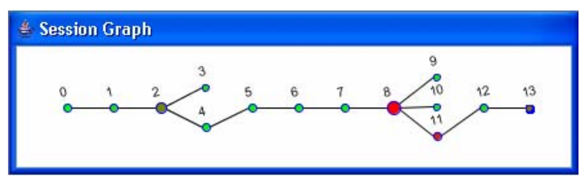


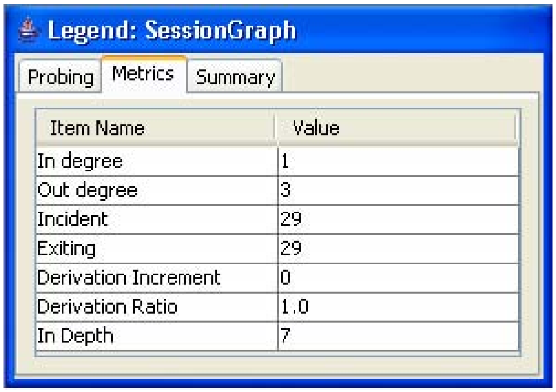
Figure 2: States of PieChart Tool in Session of Figure 1. Size of circle represents number of out-going edges.

We can draw the states of all instances of a single visualization tool or the states of all controls in a given taxonomy category (not described) or the computed process metrics. This flexibility is helpful to support users, who may remember the tool or control they used for an action quite awhile ago, and can understand its much simpler history more easily. This is what opens interesting avenues for evaluation (see Section 5.3 below.)

5. METRICS

5.1 GDE Metrics

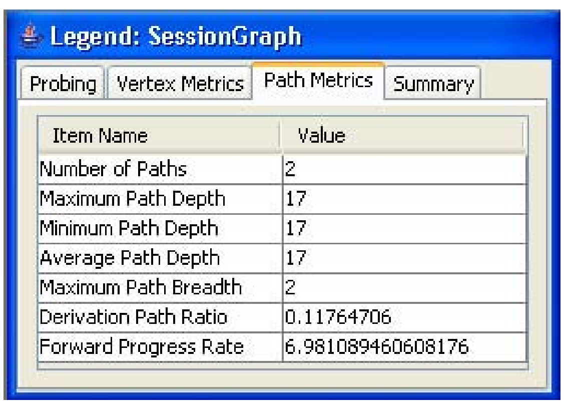
The GDE defines two kinds of metrics on the session graph: vertex metrics (Figure 3) and path metrics (Figure 4).



Item Name	Value
In degree	1
Out degree	3
Incident	29
Exiting	29
Derivation Increment	0
Derivation Ratio	1.0
In Depth	7

Figure 3: Metrics on the Selected Vertex

Vertex metrics are computed on a single node of the graph, based primarily on its in- and out-degree and total visits. Figure 3 shows the metrics the system displays for the selected (probed) vertex in the Session Graph. Vertex metrics allow nodes to be classified as "source," "terminal," "sequential," "k-sequential," "landmark," "quasi-source," and "quasi-terminal" vertices. Important states, such as "landmark vertices" that the operator repeatedly returns to, are identified for highlighting states in all the visualization tools.



Item Name	Value
Number of Paths	2
Maximum Path Depth	17
Minimum Path Depth	17
Average Path Depth	17
Maximum Path Breadth	2
Derivation Path Ratio	0.11764706
Forward Progress Rate	6.981089460608176

Figure 4: Metrics on the Entire Graph.

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.

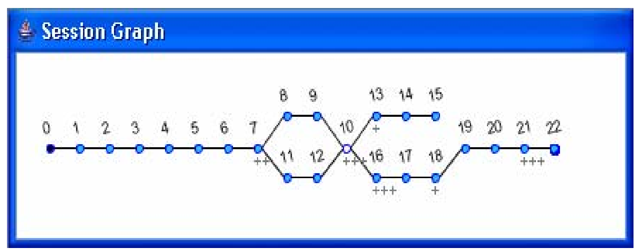


Figure 5: The GDE Forward Derivation Component Graph

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.

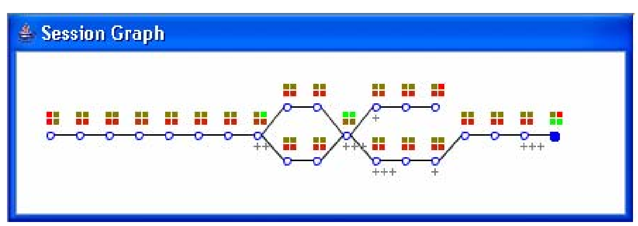


Figure 6: Heatmap representation of vertex metrics

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

[1] Gee, A.G., Li, H., Yu, M., Smrtic, M.B., Cvek, U., Goodell, H., Gupta, V., Lawrence, C., Zhou, J., Chiang, C.-H. and Grinstein, G.G., Universal Visualization Platform. in *Visualization and Data Analysis 2005*, (San Jose, California, USA, 2005), SPIE, 274-283.

[2] 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.

[3] Goodell, H., Chiang, C.-H., Kelleher, C., Baumann, A. and Grinstein, G., Collecting and Harnessing Rich Session Histories. in *10th International Conference on Information Visualisation*, (London, England, 2006), IEEE Press.

[4] Lee, J.P. and Grinstein, G.G. An Architecture for Retaining and Analyzing Visual Explorations of Databases. *Proceedings of IEEE Visualization 1995*. 101-109.

[5] Lee, J.P. A systems and process model for data exploration *Computer Science*, Dissertation, University of Massachusetts at Lowell, Lowell, Massachusetts, USA, 1998, 245.

[6] Rooden, M. J. T., Ed. (1998). Thinking About Thinking Aloud. Contemporary Ergonomics. London, Taylor and Francis.

[7] Boren, M. T. and J. Ramey (2000). "Thinking Aloud: Reconciling Theory and Practice." IEEE Transactions on Professional Communication 43(3): 261-278

[8] Bowers, V. A. (1990). Concurrent Versus Retrospective Verbal Protocols for Comparing Window Usability. Industrial Engineering and Operations Research. Blacksburg, Virginia, USA, Virginia Polytechnic Institute and State University. Doctorate thesis.

[9] Rekimoto, J., Time-Machine Computing: A Time-Centric Approach for the Information Environment. in *ACM Symposium on User Interface Software and Technology*, (1999), 45-54.