Modeling Human Performance from Visualization Interaction Histories

Steven R. Gomez*   David H. Laidlaw†

Department of Computer Science
Brown University

ABSTRACT

We present a novel framework called TOME that helps designers quantitatively evaluate interactive visualizations. TOME collects user-interaction histories from visualizations via an instrumentation library and can convert these histories into keystroke-level models (KLMs) [1] that predict task-completion times using CogTool [2, 7]. Our aim is to minimize human involvement in performance-model construction, which is notoriously time-consuming and error-prone when done manually [6]. We evaluate the framework with a brain-circuit visualization and demonstrate its use in iterative tool development.

Keywords: Quantitative evaluation, human factors, user interfaces, cognitive modeling, histories.

1 INTRODUCTION

Quantitative user studies can help visualization developers evaluate new tool designs, but these studies can be difficult to plan and carry out. Analyzing usage data on each design iteration is often prohibitively expensive. An alternate approach is to construct a predictive model of the tool's utility (e.g., task speed or accuracy for an average user) and evaluate interface changes by running the model. This paper describes a model-construction framework called TOME for building and extending these models with minimal human effort or oversight.

The contributions of this work include an early implementation of TOME, as well as a case study in brain-circuit visualization that demonstrates the framework's prediction *accuracy* for task-completion times and *usefulness* for evaluating new interaction designs. We show that performance predictions for quick (5–60 sec.) circuit query tasks average within 10% of expert performance, and we extended one TOME-generated model to evaluate a proposed feature that speeds up one task by 16%.

2 DESIGN EVALUATION BY MODELING

Interactive visualizations require design choices to be made at both the user interface/interaction level and visual representation level; while our work addresses the former, it could be coupled with perceptual modeling to evaluate design iterations more fully. We use KLM, which predicts the time it takes an expert user to execute necessary keyboard and mouse input, along with cognitive operations (e.g., "mental preparation" including eye movements to look at the display). This kind of modeling lets us predict task-completion time from a sequence of anticipated user interactions, which we find by recording many interaction histories for a task and 'voting' on which best represents the critical path. This approach prevents us from having to interpret the semantics of histories – or knowing *a priori* how users typically complete tasks, a requirement in manual KLM construction – provided each history is labeled with the task it completes.

*e-mail: steveg@cs.brown.edu
†e-mail: dhl@cs.brown.edu

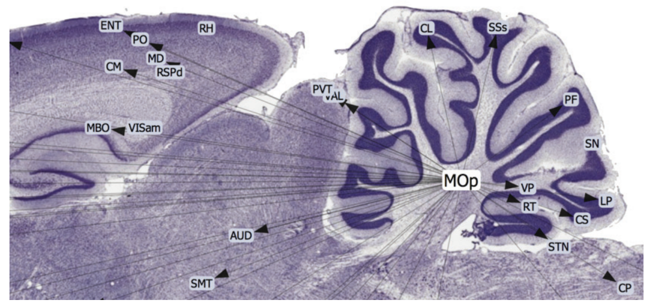

Figure 1: Interactive diagram showing efferent neural projections from an area in the rat brain. We evaluated task performance predictions and a proposed interaction with this tool (see Section 3).

TOME builds off several projects aimed at making UI modeling more accessible. Hudson *et al.*'s CRITIQUE [5] generates KLMs by having individual users demonstrate tasks on a prototype UI. John *et al.*'s CogTool [2, 7] lets users build UI prototypes for modeling-by-demonstration in a WYSIWYG editor. TOME provides a Swing-based instrumentation library for Java, compatible with popular information visualization tools like Prefuse [3]. With TOME, an instrumented visualization automatically produces interaction histories, called Tomes, as users complete their tasks. The framework aggregates a collection of Tomes that users have labeled by the same task into a canonical KLM for that task, without needing a human to inspect them individually or know the predominant strategies of users "in the wild".

While our work has incorporated interaction histories for modeling, histories have been used previously in information visualization for user experience research and to support usability, as recently demonstrated with Tableau [4].

A Pipeline for Automatic Modeling

The TOME framework functions as a pipeline that starts with instrumentation and outputs canonical task "storyboards" that can be imported and run or edited in CogTool, which is free and open-source, to get time predictions.

The ability to edit these storyboards in CogTool makes our approach more powerful than simply gathering average times from history timestamps; we can compare current UI designs against proposed changes by copying generated TOME storyboards and perturbing them in the WYSIWYG editor to reflect incremental design changes. This utilizes CogTool's ability to rapidly prototype designs as well as TOME's ability to gather some baseline model for how users currently complete a task. We describe an example of this design revision process in the next section.

3 CASE STUDY: INTERACTIVE BRAIN DIAGRAMS

We incorporated TOME into the design of an interactive visualization of the rat brain circuit, as shown in Fig. 1, in collaboration with scientists studying functional neural connectivity. We instrumented this interactive node-link diagram to collect task histories and compare empirical completion times with those predicted by the system, to establish the accuracy of these predictions. Furthermore, we show that these predictions are useful in feature design.

3.1 Experimental Design

We collected interaction histories of the brain-diagram tool from eight subjects, who were undergraduate or graduate students in computer science. The participants were split into two groups that each completed our two types of tasks (described below) with different query instances. With the consent of each participant, we recorded participant videos and screen capture for posterior analysis. Each was trained for 10-15 minutes and told to complete the following tasks as quickly and accurately as possible:

T1: *'Nearest neighbor' projections.* Given the name of a specific brain part p, *select* the two nearest parts on the map that share a projection with p (in either direction).

T2: *Map adjustment.* Given the names of two specific parts, p_1 and p_2, and a target part t, click and drag both p_1 and p_2 on top of t.

In both tasks, participants were required to interpret the graph, make decisions about what to do, and complete motor activity using the keyboard and mouse.

Each participant completed each task 25 times. The first five runs in each task tested the subject on all different brain parts so as to increase familiarity with the tool/task. The remaining 20 runs of each task were repeated with the same query; this repetition lets us estimate the average "expert" completion time (mean from runs 11–20, using a timer) for each user-task to compare to KLM, which specifically predicts expert performance. Runs 1 through 10 for each user-task are training data for our aggregation program, which constructs a storyboard/model from the most common interaction sequences.

3.2 Evaluating New Feature Designs

After gathering task histories and building TOME models, we show how these models might be extended to evaluate new features before implementing them. We used a model created for the T1 task to evaluate an unimplemented interaction called *radius select* that makes this task faster. With radius select, a user can select all brain parts within a circular area-of-interest by choosing a center and radius on the map; this interaction can be used to solve T1 quickly, without individually selecting nearby nodes.

We used CogTool to edit the T1 model built by TOME after our experiment. This amounted to adding one transition to the storyboard graphical model, triggered by a new mouse action, in the previously constructed storyboard. We simulated radius select in CogTool to produce a time prediction for experts. In fact, the prediction was 18% faster than the prediction for the original implementation, and it compelled us to implement radius select in the brain map.

3.3 Results

Fig. 2 shows results for prediction accuracy for the tasks in section 3.1. The worst error was just under 14%, on group B's T2 task. Here, we reviewed video and found that one participant repeatedly deviated from the most popular strategy that TOME automatically storyboarded, using a significantly slower interaction sequence that raised the mean expert time.

As described in section 3.2, we extended the T1/A storyboard to include the *radius select* interaction. The prediction for the T1/A task using this feature was 5.7 seconds, which is 18% faster than the original prediction (6.9 sec) and 27% faster than actual expert time (7.8 sec) from our study. Because of this predicted speed-up, we implemented this feature, tested it with 4 participants, and found it to be around 16% faster than previous expert time.

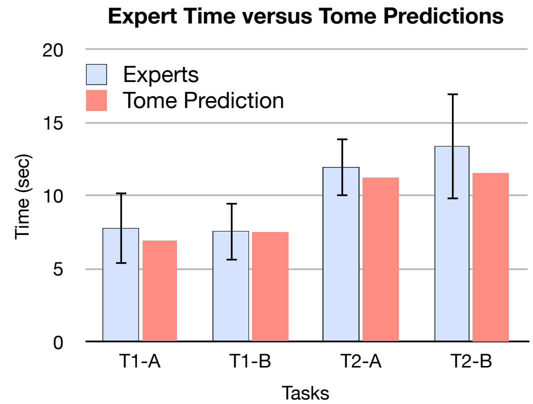

Figure 2: Average prediction error for these models is less than 10%. Error bars show ± 1 std. dev.

4 Conclusion

We have described work toward a novel framework called TOME for modeling human performance with interactive information visualizations. Unlike previous modeling tools, TOME does not require a human to know upfront how visualization users will complete tasks. We have yet to characterize failure cases for this approach beyond the limitations of KLM (e.g., tasks longer than 5 minutes). Modeling higher-level cognitive processes with minimal human expertise remains an important challenge. Still, our initial results are encouraging: for quick diagram-query tasks, we have shown that TOME generates predictions within the 20% error claimed by KLM [6] and that these models can be used to evaluate iterative designs.

Acknowledgements

This work was supported in part by NSF award IIS-10-16623 and NIH award R01-EB004155. All opinions, findings, conclusions, or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980.

[2] B. N. Harris, B. E. John, and J. Brezin. Human performance modeling for all: importing UI prototypes into CogTool. In *ACM Human Factors in Computing Systems (CHI): Extended Abstracts*, pages 3481–3486, 2010.

[3] J. Heer, S. K. Card, and J. Landay. Prefuse: A toolkit for interactive information visualization. In *ACM Human Factors in Computing Systems (CHI)*, pages 421–430, 2005.

[4] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14:1189–1196, November 2008.

[5] S. E. Hudson, B. E. John, K. Knudsen, and M. D. Byrne. A tool for creating predictive performance models from user interface demonstrations. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 93–102, 1999.

[6] B. E. John. Reducing the variability between novice modelers: results of a tool for human performance modeling produced through human-centered design. In *Proceedings of the 19th Annual Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, 2010.

[7] B. E. John, K. Prevas, D. D. Salvucci, and K. Koedinger. Predictive human performance modeling made easy. In *ACM Human Factors in Computing Systems (CHI)*, pages 455–462, 2004.