

A Tool for Enabling Scientific Exploration of Human Performance Models in HCI Education

Martin Schedlbauer

Department of Computer Science
University of Massachusetts
Lowell, MA 01854 (USA)
+1 978 934 4000

mschedlb@cs.uml.edu

Robert Pastel

Department of Computer Science
Michigan Tech. University
Houghton, MI 49931 (USA)
+1 906 487 2209

rpastel@mtu.edu

ABSTRACT

Few HCI courses offer students the opportunities for scientific research and human performance modeling. A barrier to this exploration could be the lack of experimental platforms appropriate for student use. A pedagogical tool that provides the appropriate amount of scaffolding to the students could make scientific discovery and the investigation of theoretical performance models, such as Fitts' Law, practical and accessible to student researchers. In this paper, we present a new software platform for the interactive exploration of movement time models including Fitts' law. The open source software is written in Java and provides a flexible environment for undergraduate and graduate HCI research. The patterns-based and object-orientated software design provides extensibility in regard to new models, task types, and selection modes.

Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]:
Computer Science Education

General Terms

Measurement, Design, Experimentation, Human Factors.

Keywords

Undergraduate research, human-computer interaction research, Fitts' law, human performance modeling, human factors.

1. INTRODUCTION

Many HCI courses teach user centered design using a qualitative approach [3] and neglect the study of human performance models and opportunities for the students to be creative through scientific explorations. Without instruction in fundamental human performance modeling and experience in scientific exploration, future HCI designers may be unable to create interaction techniques that will be necessary for next-generation devices and interactive environments. McCrickard *et al.* [8] and

Pastel [13] recommend that science and scientific exploration ought to be integrated into HCI courses. McCrickard *et al.* propose a unified approach to HCI education that fuses the relevant elements of design, science, and engineering into a coherent educational framework. Pastel shows how research projects can be viable alternatives to common programming projects in HCI design courses.

Scientific exploration is based on observation and analysis of the empirical measurements. Frequently, scientists design experiments and analyze their observations using statistical techniques. Given the constraints of a one semester course, there is not much time for students to implement their own software platform for experimentation. Also, if the students construct their own experimental platform they will be required to spend significant effort in debugging their platform rather than focusing on data collection and analysis. An additional obstacle for many HCI students performing experiments is their lack of skill with statistical inference and regression analysis. Nevertheless, the students can still experience many aspects of scientific exploration and conduct their own experiments if they have sufficient scaffolding that supports them through the implementation of the experiment and the associated statistical analysis [14].

Fitts' law, one of the most commonly applied human performance frameworks in HCI [5], models movement time (MT) as $MT = a + b \log_2(A/W + 1)$ where A is the distance to the target and W the width of the target; a and b are experimentally derived regression coefficients [9]. The logarithm part is the *Index of Difficulty* or *ID* of the movement. The throughput (TP) of an input device is a measure of its efficiency and is calculated as the ratio of the mean *ID* and the mean *MT* [15].

While many specialized utilities, such as the *Generalized Fitts Law Model Builder (GFLMB)* [10],[16], *WinFitts* [18], and *IDTest* [15] have been developed to capture data for evaluating the usability of input devices, few of these tools are general purpose experimental and educational platforms. This paper describes the *Movement Time Evaluator (MTE)*, an interactive research workbench for designing, executing, and analyzing Fitts-type experiments. It is an extensible tool which allows students to focus on creativity, discovery and exploration rather than programming. In addition, it provides students insight into graduate research which may encourage them to continue their education. We believe that interactive experimentation will make theoretical concepts more accessible to the students, and we hope will make them aware of the role and importance of scientific exploration in HCI.

1.1 Review of Existing Tools

Both *GFLMB* and *WinFitts* are written in C++ rather than the more accessible Java programming language. *GFLMB* runs on the Macintosh whereas *WinFitts* is for PCs running Windows. While *GFLMB* is interactively configurable and has some limited regression plotting capabilities, *WinFitts* does not contain any interactive data analysis capabilities and requires modifying the C++ code to change experimental parameters. A search of the web revealed that many HCI courses use Barton's *IDTest* program [15], which, like *MTE*, is also written in Java. While *IDTest* supports basic Fitts and steering experiments [1], it does not contain any mechanisms to interactively explore the collected data. *IDTest* writes the data to a fixed-field file format which must then be loaded into custom Java programs for analysis. This process is not conducive to interactive exploration and scientific discovery, particularly given the time limits of a semester long course.

MTE improves upon these tools in several significant ways. Experiments are interactively configurable and the configurations are saved in a sharable XML format. The experiments can be carried out via a network allowing the researcher and subject to be separated by any distance. Its data sets are saved in either XML or CSV which simplifies importing into customized programs and off-the-shelf plotting and statistical packages, such as Microsoft Excel and R, an open source statistical package [6]. Furthermore, *MTE* contains interactive plotting and basic statistical analysis allowing student researchers to collect data and "play with the data" in a single environment. The tool supports visualization of a single data set, several data sets side-by-side or overlaid, cursor movement paths, spatial variability of cursor selection end points, and cursor kinematics. Lastly, its internal object-oriented architecture and implementation in Java 5 makes extensions to and customizations of the tool relatively simple.

2. PLATFORM CAPABILITIES

MTE is based upon a distributed architecture in which a researcher controls experiments from one workstation while the subjects interact with the software on different workstations. This is particularly useful for students conducting experiments and collecting data remotely. Connectivity is through a TCP/IP socket over which serialized Java objects are employed as the message format allowing the workbench to control an experiment over any distance.

The overall platform capabilities are summarized in the UML (Unified Modeling Language) [4] use case model of Figure 1.

2.1 Configurability

Experiments are carried out in two steps: configuring followed by running. The researcher first creates an experiment setup which specifies session invariant parameters, including target extent, target shape type (oval, rectangular, moving, or soft keypad), home region placement (center, upper left corner of screen, or none), target position distribution (reciprocal, pre-programmable, or random), auditory and visual feedback preferences, number of repetitions, type of movement (point-and-click versus drag-and-drop) and the type of information to be recorded for each aiming task (cursor path and errors). The saved and sharable configuration files are used in the second

step: running the experiment. The researcher can recall a previously recorded configuration from a list of stored setups. Before each experimental session, input device type, screen dimension, probe characteristics, gain settings for the input device, as well as any other relevant *ad hoc* information are recorded. For each subject, demographic information is collected, including age, gender, height, and handedness.

MTE can be extended by adding new static as well as dynamic (moving) shape types, target position distributions, movement time models, and data export formats.

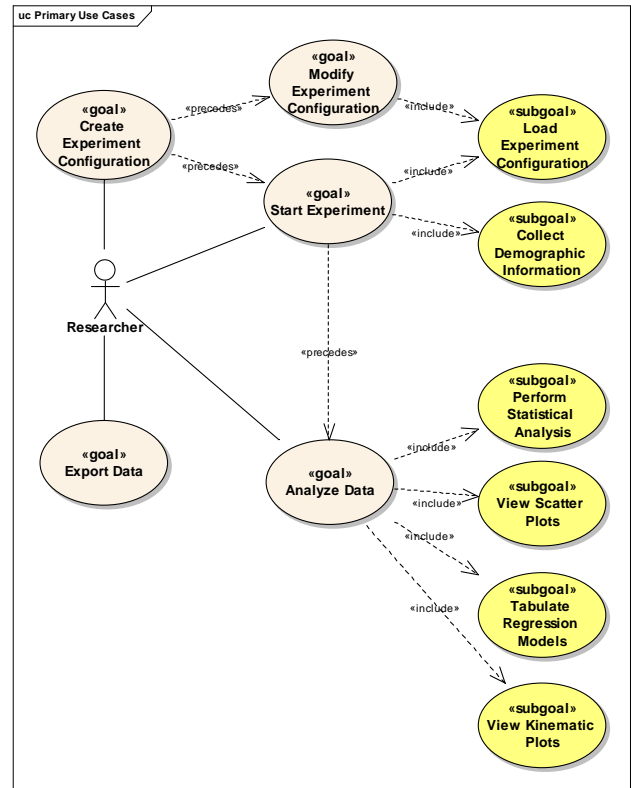


Figure 1. UML use case model for *MTE*.

2.2 Evaluation and Analysis

To assist in the rapid evaluation and interactive exploration of movement models and input devices, basic statistical analysis is built into the platform. The recorded data can be easily exported to many statistical packages for more sophisticated analysis. *MTE* supports Pearson correlation analysis and linear regression, as well as configurable scatter plots and distribution graphs. In addition, the raw data and the trajectory of the individual acquisition movements during a session can be visualized so that the movement patterns for different input devices can be studied. Lastly, a table comparing the correlation coefficients of various movement time models, including Fitts' law can be viewed.

The linear regression coefficients and the correlation coefficients can be computed either on the raw observations or averaged *MT* values across ranges of *ID*. Regression on the averaged *MT* values will attenuate the effect of outliers. A

sortable table containing collected trial data can assist in identifying and removing outliers.

2.3 Exporting to Other Tools

The tool records the experiment configuration and the data for each movement trial in a persistent and sharable XML document. While the statistical mechanisms built into *MTE* are certainly useful, they are limited. *MTE*'s built-in statistical methods are easily augmented by specialized statistics packages, such as R [6]. Data is transferred to these external packages by either copying data to the clipboard or saving it into a comma-separated values (CSV) file. Although the plotting capabilities of *MTE* are useful for interactive exploration, they are not configurable enough for publication. Both R and Microsoft Excel offer better support for producing publishable plots and analytical results.

2.4 Availability

The *MTE* platform is available under the GNU Public License (GPL) as open source software. An executable version with an installation wizard that runs under Java 5 or later as well as the full source code can be downloaded from the web¹, allowing for modification and extension by students and researchers. The software has been tested on Windows 98, Windows 2000/NT/XP, Linux, and Mac OS X, all running Sun Java 5 [17].

3. CASE STUDY

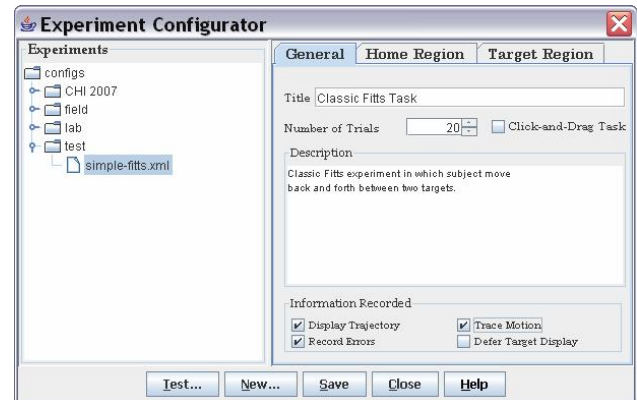
This section demonstrates how *MTE* was used by a team of graduate students to evaluate the performance characteristics of three input devices: touch screen, mouse and trackball.

3.1 Configuration

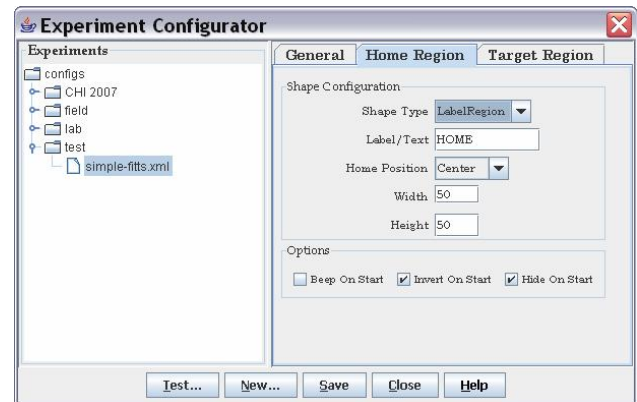
For the device evaluation, a two-dimensional pointing experiment using circular targets was configured. Twenty movement trials for four different target extents (15, 35, 45, and 60 pixels) were specified. The amplitude of the movement was randomly varied, but the same positions were displayed in the same order for all trials. So, the tool's *FixedRandomDistribution* positioning strategy was chosen. This resulted in a set of random target positions that was the same in all runs of the experiment. Other distribution functions consist of random placement with a random seed, reciprocal placement where the targets move back and forth between the left and the right side of the home position, and a static distribution where targets are displayed at fixed and pre-programmable positions. The order of display can be randomized to mitigate the effects of learning in repeated trials.

In both experiments, the subjects had to first tap a home region before moving to the target, simulating a point-and-click task rather than a drag-and-drop task. The researcher can choose whether to display the target immediately or only once the home region is tapped. Deferring the display of the target implies that the recorded target acquisition time includes both the movement and the reaction time.

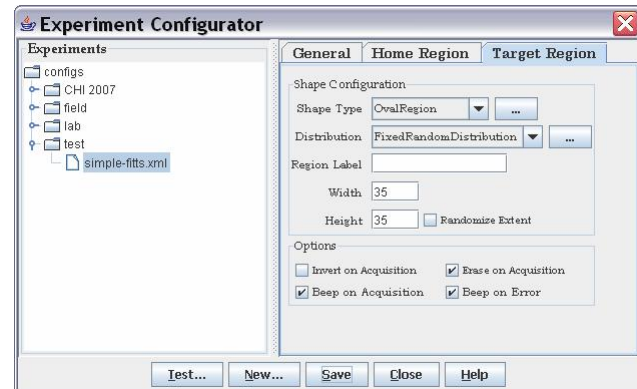
The steps required to configure the experiment are illustrated in Figure 2. Figure 3 shows the starting of an experiment trial and how additional *session specific* data is collected.



Step 1: Specify general parameters. The cursor path is recorded and displayed on the screen. Error selections are also recorded.



Step 2: Specify a square home region with width 50 pixels centered on the screen. The region is hidden upon selection to signal the start of the timing.



Step 3: Specify an oval target of width 35 pixels randomly placed on the screen. The target is erased upon selection to signal the end of the timing. Auditory feedback is provided for correct and incorrect selections.

Figure 2. Configuration sequence of a Fitts experiment.

¹ Available at <http://www.cs.uml.edu/~mschedlb/mte>.

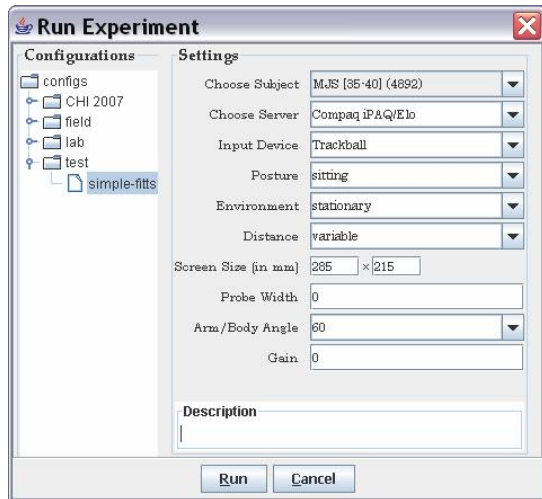


Figure 3. Collection of session-specific data prior to starting an experiment trial using a stored configuration file.

Figure 4 shows a partial screen shot of an interaction with the experiment. The experiment has been set up to display as well as record the actual trajectory of the movement for the mouse and trackball, along with the momentary speed at each recorded sample. This provides detailed acceleration information to the researcher and allows for comparison of cursor trajectories, movement accuracy, and cursor kinematics.

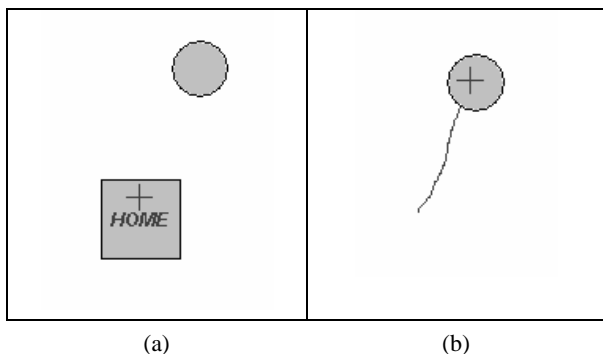


Figure 4. Partial screen capture of a bivariate pointing task with cursor trajectory: (a) shows home region before timing starts and (b) shows target acquisition after the home region is selected. The cursor path is displayed and recorded.

3.2 Exploration and Evaluation

After the raw data was gathered, the data sets collected for each subject were merged into a single combined data set. Data sets can be explored in their raw form or in a “scrubbed” form where the data is binned (grouped and averaged) and outliers are removed. In this evaluation, the averaged data was used, but no outliers were removed.

Figure 5 shows a scatter plot of the averaged data for the trackball trials along with the correlation coefficient and linear regression equation for Fitts’ law. Movement time (*MT*) is along the Y axis and the Index of Difficulty (*ID*) is along the X axis. The tool allows any recorded parameter to be plotted against any other recorded parameter, *e.g.*, amplitude or target angle against

movement time. To facilitate the comparison of different experiments, multiple data sets can be plotted side-by-side or overlaid. Figure 6 shows an overlaid scatter plot that contains the average movement times for both mouse and trackball. The trackball movement times were markedly slower; consequently the trackball data is above the mouse data.

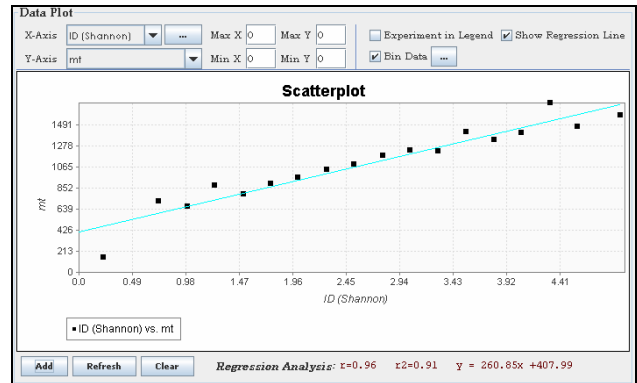


Figure 5. Screen capture of regression plot of ID versus MT with correlation coefficient (Pearson moment) and linear regression equation.

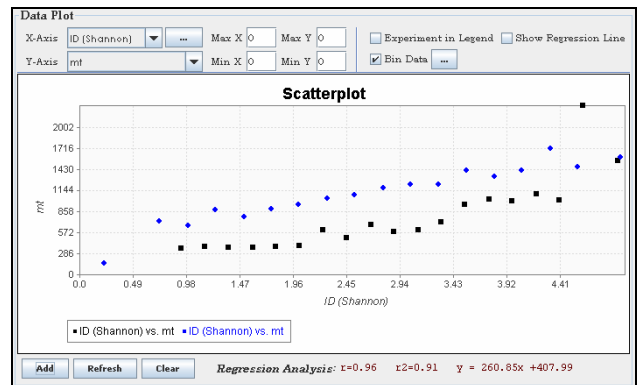


Figure 6. Screen capture of scatter plot showing ID versus MT for mouse (squares) and trackball (diamonds).

The movement model panel is shown in Figure 7. The panel summarizes the mouse correlation results and throughput calculations for several movement time models, including those by Fitts, Welford, MacKenzie [9], Kvalseth [7], Meyer *et al.* [11], and Accot & Zhai [2]. It shows the correlation coefficient (*R*) and the coefficient of determination (*R*²) for each model using the Euclidean distance between the targets as the amplitude as well as the length of the cursor trajectory.

The movement model panel displays minimum, maximum, and mean *ID* and *MT*. The panel also shows the throughput calculated using both the ratio of *ID* over *MT* and the inverse of the slope of the regression line. Both forms of throughput are in use, although the latter is more common [19].

| | Simple | Fitts | Welford | Shannon | Meyer et al. | Kvalseth | Accot-Zhai | ID^2 |
|------------|----------|----------|----------|----------|--------------|----------|------------|----------|
| R (A, W) | 0.82 | 0.95 | 0.98 | 0.96 | 0.92 | 0.82 | 0.95 | 0.88 |
| R (D, W) | 0.90 | 0.94 | 0.96 | 0.96 | 0.95 | 0.90 | 0.95 | 0.85 |
| R2 (A, W) | 0.67 | 0.91 | 0.95 | 0.91 | 0.85 | 0.67 | 0.90 | 0.77 |
| R2 (D, W) | 0.82 | 0.88 | 0.91 | 0.91 | 0.90 | 0.82 | 0.91 | 0.72 |
| 1/b | 48.34 | 4.89 | 4.31 | 3.83 | 4.48 | 48.34 | 3.97 | 5.82 |
| TP-A (bps) | 8.29 | 3.18 | 2.46 | 2.55 | 2.41 | 8.29 | 2.62 | 1.57 |
| TP-D (bps) | 10.53 | 3.45 | 2.71 | 2.79 | 2.71 | 10.53 | 2.86 | 1.75 |
| Mean (ID) | 10.18 | 3.90 | 3.02 | 3.13 | 2.96 | 10.18 | 3.22 | 1.92 |
| Max (ID) | 35.53 | 6.15 | 5.17 | 5.19 | 5.96 | 35.53 | 5.28 | 4.64 |
| Min (ID) | 0.07 | 0.01 | -0.82 | 0.09 | 0.26 | 0.07 | 0.10 | 0.06 |
| Mean (MT) | 1,227.90 | 1,227.90 | 1,227.90 | 1,227.90 | 1,227.90 | 1,227.90 | 1,227.90 | 1,227.90 |
| Max (MT) | 5,128.00 | 5,128.00 | 5,128.00 | 5,128.00 | 5,128.00 | 5,128.00 | 5,128.00 | 5,128.00 |
| Min (MT) | 151.00 | 151.00 | 151.00 | 151.00 | 151.00 | 151.00 | 151.00 | 151.00 |
| Intercept | 1,021 | 444 | 531 | 408 | 531 | 1,021 | 416 | 869 |
| Slope | 21 | 205 | 232 | 261 | 223 | 21 | 252 | 172 |

Figure 7. Screen capture of the movement model panel comparing the fit for several movement time models.

To better understand the movement characteristics of the two different indirect input devices, the graduate students investigated the trajectory paths for each target selection. The plot in Figure 8 contains the starting position of the movement, the position at which the target was successfully acquired, the direct path to the target, and the actual path traveled. *MTE* can calculate the standard deviation of the cursor coordinates from the direct (and ideal) path to the target for each target acquisition trial. This calculation is referred to as *movement variability*. A comparison of the movement variability using a *t*-test (executed in R) for the two indirect input devices (mouse and trackball) showed a significant difference ($p < 0.05$), implying that trackball movement is more “jittery” and less direct, accounting partially for the increased movement time.

A different view of a single movement is shown in Figure 9 as a plot of the momentary speed along the trajectory path from the home position to the target. Note how the movement is initially accelerating, and then slows down as the cursor moves toward the target. This plot provides a visualization of the corrective submovement model for human motor control postulated by Meyer *et al.* [11].

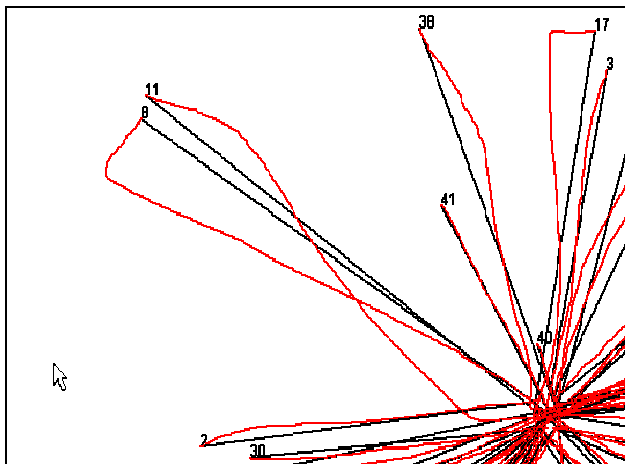


Figure 8. Screen capture of trackball trajectories. The straight line represents the direct path to the target acquisition point.

Figure 10 shows selection end points for touch input and for the trackball. The touch selections (shown as filled circles) appear to be less dispersed (*i.e.*, more densely clustered) than the selection points for the trackball (marked by open circles). This

visualization prompted a student to investigate whether selection point spatial variability is an indicator of device accuracy that might be more useful than error rate. The student exported the data to R and used a paired *t*-test to test the hypothesis. The differences in the spatial variability were indeed significant ($p < 0.05$), implying that trackball selection is less accurate than touch selection.

Without the interactive and explorative nature of *MTE*, the theoretical model of Fitts is not as easily comprehended. The plots and graphs provide a visualization of movement and selection characteristics of the input devices. The immediate feedback prompts students to formulate and test new hypotheses, sparking creative analysis.

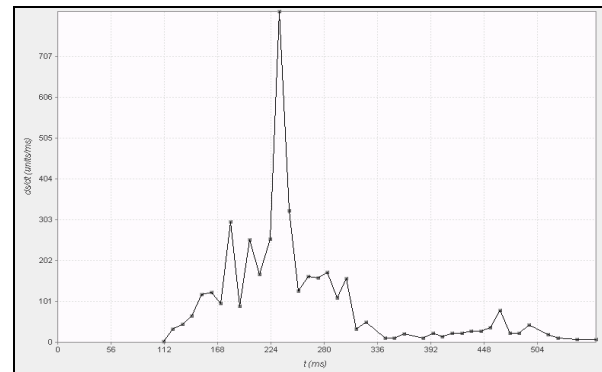


Figure 9. Graph of time since start of movement versus speed.

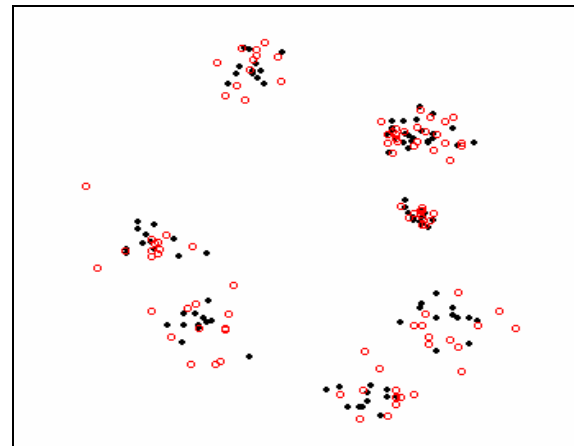


Figure 10. Spatial variability of selection end points for two different input devices. The trackball selections are shown with open circles, while the touch selections are shown with filled circles.

4. CURRICULUM INTEGRATION

MTE can be incorporated into an undergraduate or graduate HCI course during a module covering human performance modeling and Fitts’ law. Two one-hour lectures are generally sufficient to demonstrate the use of *MTE*. After the short introduction, students can then create and run experiments for an assignment validating Fitts’ law for different devices or varying interaction modes. An advanced project would be a study of atypical

populations, *e.g.*, elderly or handicapped users. Alternatively, it can be used as a basis for a research project in which students formulate a hypothesis, collect data, and validate the hypothesis through statistical analysis of the data. Finally, it can also be used as a programming project where students extend the tool to support a new task type, new analysis mechanism, new visualization techniques, or new performance model.

4.1 Student Feedback

Students who used the tool felt that a simple installation procedure is critical rather than compiling the code and installing external libraries themselves. For this reason, *MTE* is distributed to students and researchers as a self-installing package. Students also reported that *MTE* greatly simplified running experiments by providing an intuitive and consistent user interface that balanced “design for learning” and “design for usability.” They also found the process of testing, batching results, and analyzing data to be intuitive. Also ranked high on the list of important features was the ability to run experiments remotely. Undergraduate students in particular appreciated the simplicity of replicating many published studies on Fitts’ law. The undergraduate students would spontaneously refine the experiments, formulating new hypotheses, and conducting follow-on experiments leading to new interactive insights.

5. CONCLUSION

In our experience, student researchers prefer packaged tools for experimenting rather than having to program their own test platform. This increases the students’ focus on discovery and scientific exploration and provides students with critical research skills. For example, a student who used the tool decided to pursue graduate studies in HCI, because he enjoyed exploring and researching performance models. *MTE* was also used by a graduate student in a research project that evaluated a novel input device for use in assistive technology [12]. He reported that without *MTE* he would not have conducted the experiments because the development of the necessary experimental software would have been far too time-consuming.

In summary, *MTE* is well-suited to discovery-based HCI education; it provides scaffolding and an interactive means for teaching students about modeling. Because *MTE* encourages exploration, it is a valuable teaching and research tool that prompts students to think more creatively about Human-Computer Interaction.

6. ACKNOWLEDGMENTS

The authors wish to thank Dr. Jesse Heines for his comments on early draft of this paper. The authors also thank the many volunteers who participated in the experiments as well as Mark Micire, Chris Granz, Kate Tsui, and the many other students who have used *MTE* and provided valuable feedback.

7. REFERENCES

- [1] Accot, J, & Zhai, S. Beyond Fitts’ Law: Models for trajectory-based HCI tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1997)* (Atlanta, GA, USA, 1997) ACM Press, 1997, 295-302.
- [2] Accot, J, Zhai, S. Refining Fitts’ law models for bivariate pointing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2003)* (Ft. Lauderdale, FL, USA, April 2003) ACM Press, 2003, 193-200.
- [3] Barnes, J., McCracken, D., Bryant, R., and Reiser, S. Teaching Human-Computer Interaction: Reports from the Trenches. In *Proceedings of the SIGCSE Technical Symposium on Computer Science Education (SIGCSE ’03)* (Reno, Nevada, USA, February 19-23, 2003) ACM Press, New York, NY, 2003, 125-126.
- [4] Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Modeling Language User Guide, 2nd Edition*, Addison-Wesley, Boston, MA, 2005.
- [5] Carroll, J. (Ed.) *HCI models, theories, and frameworks: Toward a multidisciplinary science*, Morgan Kaufmann Publishers, San Francisco, CA, 2003.
- [6] Gentleman, R., Ihaka, R. <http://www.r-project.org/>.
- [7] Kvålseth, T. An Alternative to Fitts’ Law. *Bulletin of the Psychonomic Society*, 16(5), 1980, 371-3.
- [8] McCrickard, D., Chewar, C., and Somervell, J. Design, Science, and Engineering Topics? Teaching HCI with a Unified Method. In *Proceedings of the SIGCSE Technical Symposium on Computer Science Education (SIGCSE ’04)* (Norfolk, Virginia, USA, March 3-7, 2004) ACM Press, New York, NY, 2003, 31-35.
- [9] MacKenzie, S. Movement time predictions in human-computer interfaces. In *Readings in Human-Computer Interaction, 2nd Edition*, Morgan Kaufman, Los Altos, CA, 1995, 483-493.
- [10] MacKenzie, S., Buxton, B. A tool for the rapid evaluation of input devices using Fitts’ law models. *SIGCHI Bulletin*, 1993, 25(3), 58-63.
- [11] Meyer, D., Abrams, R., Kornblum, S., Wright, C., & Smith, J. Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements. *Psychological Review*, 95:3, 1988, 340-370.
- [12] Micire, M., Schedlbauer, M., Yanco, H. *Horizontal Selection: An Evaluation of a Digital Tabletop Device*. Technical Report, University of Massachusetts Lowell, November 2006. (submitted for publication)
- [13] Pastel, R. Integrating Science and Research in an HCI Design Course. In *Proceedings of the SIGCSE Technical Symposium on Computer Science Education (SIGCSE ’05)* (St. Louis, Missouri, USA, February 23-27, 2005) ACM Press, New York, NY, 2003, 31-35.
- [14] Quintana, C., Krajcik, J. and Soloway, E., A Case Study to Distill Structural Scaffolding Guidelines for Scaffolded Software Environments, In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ’02)* (Minneapolis, Minnesota, USA, 2002), ACM Press, New York, NY, 2003, 81-88.
- [15] Smith, B. *IDTest Cursor Motion Test Application*, 1999, <http://www.almaden.ibm.com/u/basmith/testapp.html>.
- [16] Soukoreff, W., and MacKenzie, I. S. Generalized Fitts’ Law Model Builder. In *Proceedings of the ACM Conference on*

Human Factors in Computing Systems (CHI '95) (Denver, Colorado, USA, 1995) ACM Press, New York, NY, 1995, 113-114.

- [17] Sun Microsystems, Java SE, Version 5, 2006, Available at <http://java.sun.com/javase/downloads/index.jsp>.
- [18] University of Oregon HCI Research Laboratory. *WinFitts: Two-dimensional Fitts experiments on Win32*. Available at <http://www.cs.uoregon.edu/research/hci/research>.
- [19] Zhai, S. Characterizing computer input with Fitts' law parameters – the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, 61, 6, 791-809.