Novice Use of a Predictive Human Performance Modeling Tool to Produce UI Recommendations

Kyung Wha Hong and Robert St. Amant
Department of Computer Science
North Carolina State University, Raleigh, NC 27603
khong@ncsu.edu, stamant@ncsu.edu

ABSTRACT
This note describes two studies of the use of a performance modeling tool, CogTool, for making recommendations to improve a user interface. The first study replicates findings by Bonnie John [7]: the rates at which novice modelers made correct recommendations (88.1%) and supported them (68.2%) are close to the values in John’s study (91.7% and 75.1%, respectively). A follow-on study of novice modelers on the same task without CogTool produced significantly lower values. CogTool improves the UI design recommendations made by novices.

Author Keywords
CogTool; usability analysis; interface design.

ACM Classification Keywords
H.5.2 User Interfaces: Evaluation/methodology.

General Terms
Human Factors; Design.

INTRODUCTION
CogTool [6] is a cognitive modeling tool for HCI; it predicts human performance for storyboarded user interfaces. In the past few years it has been applied in real-world software development projects to good effect [1]. John [7] reports a study in 2011 of 100 novice modelers who used CogTool to produce recommendations for improving two Web sites. Modelers show significant consistency in their results, and the majority of their recommendations are well supported with quantitative justifications based on CogTool models. These results are important: modeling tools are typically viewed as helpful mainly to expert modelers in understanding user performance; John demonstrates that CogTool can be easily learned and applied by those new to modeling and relatively new to interface design.

This note describes a replication of John’s study. Wilson et al. [8] advocate for replication in HCI, to ensure that past results generalize, to check assumptions about prior work, and to build new findings on solid ground. The first contribution of this note is the finding that John’s results generalize beyond the sample of novice modelers at CMU, taught to use CogTool by one of its developers. Novice modelers at a different university, taught by an instructor with light experience using CogTool, produced and justified design recommendations for improving a specific user interface with similar accuracy and consistency. The implication is that learning of CogTool can scale, knowledge we believe is important for shaping future generations of HCI practitioners in the classroom.

CogTool shares the goals of other task modeling techniques for HCI (e.g., the Keystroke Level Model [2]), but we are aware of no empirical comparison between CogTool and these techniques for making design recommendations. A second contribution is a follow-on study in which novice modelers, without knowledge of CogTool but with the same exposure as in Study 1 to the KLM and other techniques, recommended improvements to the same user interface. Relatively large differences in performance were observed. In the remainder of this note we describe the two studies and discuss their implications.

METHOD
Undergraduate computer science students in an HCI course participated in the studies. The course is a conventional survey of topics in HCI, using The Resonant Interface [4] as the textbook. At the time of each study, modeling topics in class had covered Fitts’ Law, KLM, GOMS, and modeling for HCI in general (three 1.25-hour lectures in all). Students (hereafter “modelers”) submitted models as part of a homework assignment, worth 5% of their course grade.

In both studies, modelers were given an explicit task to perform on a Web site, as shown in Figure 1. We chose a simple task comparable to John’s for interpretability of our satisfaction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2014, April 26 - May 01 2014, Toronto, ON, Canada
Copyright 2014 ACM 978-1-4503-2473-1/14/04...$15.00.
http://dx.doi.org/10.1145/2556288.2556972

1. Visit the Web page http://www.amazon.com. (If you are automatically logged into your Amazon account, log out, then reload the page.)
2. Click in the Search text box.
3. Type “kinect” (without quotation marks) in the Search text box, and press return.
4. Click on the top link that is shown in the search results; for me this is “Kinect Sensor with Kinect Adventures and Gunstringer Token Code.”
5. Click on the Add to Cart button on the right.

Figure 1. Task instructions given to novice modelers
Determine three potential improvements to the usability of the Web site, for this task, based on the information in the visualization. For each of your improvements,

1. Give a short phrase that identifies the improvement.
2. Optional: If the phrase doesn’t entirely specify what the change to the Web site should be, explain it in more detail.
3. Give a screenshot of the visualization, annotated to show how the improvement would change performance (e.g., a circle around some pattern).
4. In words, summarize the pattern you’ve identified in the visualization.
5. Give the time savings, in seconds and milliseconds, that your improvement would produce.

Figure 2. Modeling instructions given to novice modelers

If our results did not match, we would have an additional explanatory factor for the differences. Our interest is in the generalizability of CogTool across populations of users; we must leave generalization to more complex tasks for future work.

Modelers were also given instructions for building a model for the task and making three recommendations for improving the interface. Additional instructions, elided here, directed modelers to carry out the task by hand first, to be as specific as possible in their recommendations, and to remember that (a) the goal was to model expert behavior and (b) not all potential changes, such as reducing visual clutter or changing the color scheme, would plausibly improve expert behavior, and such changes were not amenable to analysis using the techniques the modelers were to rely on.

Figure 2 gives the modeling instructions for Study 1: modelers were to build their models in CogTool and support their recommendations with annotations of a CogTool timeline visualization. Study 1 replicates John’s as follows: the participants were new to modeling and CogTool, they were given a step-by-step description of a task, and they completed their modeling and recommendation activities within one week. The instructions were equivalent to John’s, though more explicit in describing the required structure for submissions. There were differences. The participants were all computer science majors rather than HCI majors with a mix of backgrounds, and they were asked to analyze a different Web site from the ones used by John. The introduction to CogTool, including a demonstration, took an additional half hour of class time. The introduction was given by someone not on the CogTool development team, and unlike John’s study no hands-on working session followed during which questions could be answered. (A few procedural questions were answered via email.) Otherwise Study 1 was as close to John’s as practicable.

When we attribute performance to a specific influence—here, CogTool—we must ask about performance in its absence. We thus conducted a follow-on study one semester later in the same course. The conditions in Study 2 were the same as in Study 1, except that the modeling task was given to students before the introduction of CogTool. (They used CogTool later in the course for their project work.) The modeling instructions for Study 2 gave more generic guidelines, without mention of CogTool: Base your structured description on the material in Chapter 7 and our discussion in class: the Model Human Processor, with its visual, motor, and cognitive capabilities; the Keystroke Level Model; various GOMS techniques. Modelers worked without specialized tools, mostly relying on Fitts’ Law and KLM. Aside from the use of CogTool, Study 1 and Study 2 are effectively identical, with respect to the task and the background of the participants.

STUDY 1: RECOMMENDATIONS WITH COGTOOL

In Study 1, 53 modelers submitted recommendations. Two submissions were produced without using CogTool and are not included in our analysis. Of the remaining 51, two modelers provided only two recommendations rather than three, giving 151 recommendations in all.

Two evaluators independently created categorizations over the aggregate set of recommendations. The evaluators then worked together to agree on a specific category for each recommendation. As shown in Figure 3, the most common recommendations were to place an additional “Add to cart” button beside each product and to make targets for mouse clicks larger or to place them closer together.

Following John, the recommendations were judged with respect to being “correct” (whether they would reduce the predicted duration of the task) and “well supported” (by reference to a visualization produced by CogTool). The evaluators judged 68.2% of the recommendations correct and well supported (c+ws), below John’s 75.1%. An additional 19.9% were correct but “vague or inarticulately argued,” in most cases because the explanation did not refer to the model (cf. John’s 16.6%). The remainder were incorrect. In total, 88.1% of the recommendations were judged to be correct, if not necessarily well supported (cf. 91.7%).

Figure 3 shows all of the correct and well-supported recommendations given by modelers; no others were identified. All but one modeler (98%; cf. John’s 100%) gave at least one correct recommendation. 42 of the 51 gave at least one correct and well-supported recommendation (82%; cf. John’s 96%—possibly due to the abbreviated instructions for CogTool in Study 1). Fourteen modelers gave one or more incorrect recommendations (27%; cf. John’s 21%).

---

1 Our emphasis is not on the categorization of novice design recommendations, but we note the challenge of consistent categorization of usability improvements based on textual descriptions and CogTool visualizations, working without an a priori set. The two evaluators (with both research and practical experience in HCI and usability) differed in the number and assignment of categories they identified, due to ambiguities or combinations of categories in the modelers’ descriptions. There was initial agreement on 74% of the recommendations, with Kappa = 0.693.
These represent poorer performance than John’s numbers, but most measures are surprisingly close.

Figure 3 also shows the median duration and interquartile range of the time savings calculated by the modelers, per category of recommendation. The total time for the task is more than ten seconds, and most of the recommendations would each save half a second or more. We find it notable that novice modelers can justify recommendations for improving a professionally designed interface in wide use.

Variability in estimates of time savings (in the IQR column) is inevitable, due to the coarseness of categorization. For example, changing the size or location of a target icon can reduce duration, but how much depends on the new size or location; Search autocomplete and Incremental search can reduce typing time, but the reduction depends on how many keystrokes are saved. Some but not all modelers modified their models for these two design changes; others estimated the savings by subtracting some number of keystrokes. The latter approach appeared to be common, but it will not always produce accurate results. Incremental search savings are inflated because most modelers did not include the required shift of visual attention to a dynamically changing set of products. Even for more straightforward recommendations, such as Autofocus in search box, modelers differed in whether the Think operator should be included in the savings. As long as the recommendation was correct and the annotation included the relevant cognitive/visual/motor components of the task, it was considered well supported.

One category of interest not included in Figure 3 is Remove extraneous content. Many modelers observed that removing clutter could improve performance by allowing for target elements to be closer together (leading to a Closer/larger targets categorization), but other modelers recommended reducing clutter based on incorrect arguments (e.g., that experts are distracted by a cluttered visual display, or that thinking time is spent processing screen information not included in a model). These were considered incorrect.

In general, incorrect recommendations were of two types: within the scope of the task but not likely to improve an expert user’s performance (e.g., removing ads or changing button colors—8% of all recommendations); or out of scope, without regard for expertise (e.g., providing more results to scroll through—4% of all recommendations).

We evaluated the consistency of the recommendations with the Any-Two Agreement (A2A) statistic [5]. A2A is the number of common problems found by two modelers, divided by the total number of problems found by those two modelers, averaged over all modelers. John’s A2A value of 34% puts her study above 9 of the 12 studies in Hertzum and Jacobsen’s survey [5]; our A2A value of 30% is above 8. Study 1 adds evidence to the view that CogTool may reduce the evaluator effect by providing a model-based grounding for recommendations [7].

Two weeks after the completion of the modeling exercise a brief, informal survey was given. 47 modelers submitted free-form text answers to questions about their use of CogTool. (The survey was anonymous to allow the question, Did you follow the instructions?; all respondents answered yes.) The most informative answers were to the question, Describe in a few sentences how CogTool fit into the process of your making recommendations for design improvements. John has demonstrated that novice modelers can produce useful results with CogTool; our survey provides initial clues about how they integrate modeling into their analysis, as well as how CogTool improves the process.

Our informal analysis of the responses showed three categories of interest. The most common answer, from 26 modelers (51%), described using CogTool to identify appropriate areas for design changes. “I would use CogTool to see which areas of an application take the longest for a user to move through.” “I looked at the model that CogTool presented and looked for areas where there were actions canceling each other out, such as moving a hand to the mouse and then back… and tried to understand how to eliminate some of those steps.” Most then described using CogTool to justify their recommendations in quantitative terms.

Nine modelers (18%) also mentioned making comparisons between models, to evaluate the differences between interface designs. “I then looked at side-by-side comparisons… to see where differences existed.”

Thirteen modelers (25%) only mentioned using CogTool to validate design decisions they had thought of on their own: “It helped give me proof that my suggestions would actually help by providing estimated times.”.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th># Modelers (%)</th>
<th>Median</th>
<th>IQR</th>
<th># Modelers (%)</th>
<th>Median</th>
<th>IQR</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Add to cart” button for each product</td>
<td>25 (49%)</td>
<td>1.93</td>
<td>0.22</td>
<td>18 (32%)</td>
<td>2.33</td>
<td>1.30</td>
</tr>
<tr>
<td>Closer/larger targets</td>
<td>22 (43%)</td>
<td>0.18</td>
<td>0.35</td>
<td>8 (5%)</td>
<td>0.70</td>
<td>0.86</td>
</tr>
<tr>
<td>Incremental search, per-keystroke updates</td>
<td>17 (33%)</td>
<td>1.67</td>
<td>0.72</td>
<td>10 (7%)</td>
<td>0.75</td>
<td>1.83</td>
</tr>
<tr>
<td>Autofocus in search box</td>
<td>15 (29%)</td>
<td>2.54</td>
<td>1.25</td>
<td>20 (13%)</td>
<td>1.64</td>
<td>1.23</td>
</tr>
<tr>
<td>“I feel lucky” button</td>
<td>4 (8%)</td>
<td>1.89</td>
<td>1.00</td>
<td>5 (3%)</td>
<td>2.01</td>
<td>2.38</td>
</tr>
<tr>
<td>Keyboard navigation versus mouse clicks</td>
<td>4 (8%)</td>
<td>0.57</td>
<td>3.40</td>
<td>3 (2%)</td>
<td>0.93</td>
<td>3.57</td>
</tr>
<tr>
<td>Search autocomplete</td>
<td>4 (8%)</td>
<td>0.70</td>
<td>0.55</td>
<td>2 (1%)</td>
<td>0.26</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Figure 3. Correct and well-supported recommendations, with median and interquartile range of time savings in seconds.
Figure 4. Study 1 versus Study 2 performance per modeler as tail distribution functions: $c\%$ shows the percent of modelers giving $k$ or more correct recommendations, $c + ws\%$ $k$ or more correct and well-supported recommendations.

### STUDY 2: COMPARATIVE ANALYSIS

For Study 2, in which modelers did not use CogTool, 50 modelers submitted 150 recommendations in total. The same evaluators followed the same procedure as in Study 1 to categorize recommendations, except using references to the modeling techniques above rather than to CogTool in their evaluation. Figure 3 shows that the correct and well-supported recommendations in Study 2 are comparable to those of Study 1 in the numerical distribution of categories and estimates of time savings. But the totals were considerably lower. In Study 1, 88% of recommendations were correct, versus 63% in Study 2; for correct and well-supported recommendations the values were 68% and 43%.

Figure 4 shows the tail distributions for the count of correct/well-supported recommendations, aggregated over modelers; that is, 35/51 or 69% of modelers in Study 1 gave three correct recommendations, and so forth. A complete statistical analysis is beyond the scope of this note, but a two-sample Kolmogorov-Smirnov exact test shows a significant difference between Study 1 and Study 2: $D = 0.321$, $p < 0.01$. (Treating the counts as continuous data, we can ask about the difference between the mean number of correct recommendations per modeler, 2.6 in Study 1 versus 1.9 in Study 2: $t(99) = 3.97, p < 0.01$.) Analysis of correct and well-supported recommendations shows the same result. In other words, performance in these studies is not determined purely by the task and the background knowledge of modelers; use of CogTool is a significant factor.

A2A in Study 2 is 20%. The drop from 30% in Study 1 is not unexpected, given the greater variation in modeling procedures. No new correct and well-supported recommendations were identified, but there was a wider range of incorrect recommendations. These fell into the same general classes as in Study 1: changes affecting performance outside the scope of the task (e.g., multiple item selection or filtering capabilities—11% of all recommendations), and not focusing on expert users as specified in the instructions (25% of all recommendations). The latter is not surprising. In an HCI survey course, the clearest illustrations of usability issues are often based on the challenges that novice users face in complex interfaces; CogTool currently targets expert users. The former category of incorrect recommendations is more surprising, though not in retrospect. CogTool also appears to help novice modelers remain aware of the boundaries of a specific task when attempting to improve user performance by changing the UI design.

### CONCLUSION

John's experience teaching CogTool to HCI majors at CMU extends to computer science majors at a different university (without a formal HCI program), and detailed knowledge of CogTool is not needed for adequate instruction. This is promising news for practical modeling, recognized by some of the novice modelers: "I would never have thought about these areas taking up users’ time, so CogTool did help…"

Existing research provides limited information about how modeling can best be integrated into interface design tasks (e.g., [1, 3]), but the strategies used in Study 1 are among those typically recommended by experienced HCI modelers. Notably, the novices were able to find effective ways to use CogTool without explicit guidance, to produce correct, well-supported recommendations.

Study 2 indicates that CogTool is a decisive factor in helping novice modelers produce good design recommendations—it was not because the task was too easy. With CogTool, recommendations were much more often correct and well supported, and the proportion of modelers who made such recommendations was higher. CogTool appears to help novice modelers to focus their recommendations on those relevant to a specific class of users, within the boundaries of specified tasks. Further exploration of how and why, as well as implications for professional UI development, remain for future work.

### REFERENCES