Measuring the Gulf of Evaluation in Display-Based HCI

Muneo Kitajima
National Institute of Bioscience and Human-Technology
1-1 Higashi Tsukuba Ibaraki 305, JAPAN
e-mail: kitajima@nibh.go.jp

Peter G. Polson
Institute of Cognitive Science
University of Colorado, Boulder, CO 80309-0345
e-mail: ppolson@psych.colorado.edu

DEVELOPMENT OF A COMPUTER-BASED COGNITIVE MODEL OF DISPLAY-BASED HCI

Kitajima and Polson have developed a comrehension-based computational model of display-based human-computer interaction (Kitajima and Polson, 1992; 1994; to appear). The model elaborates Hutchins, Holland, and Norman's (1986) action theory framework which consists of the following four basic components:

(1) *goals* representing what the user wants to accomplish,

(2) *a task environment* which is the world that reacts to the user's actions and generates new responses by modifying the display,

(3) *the stage of evaluation* comprised of the processes that evaluate and interpret the display, and

(4) *the stage of execution* comprised of the processes that select and execute actions that affect the world.

Our model of the Hutchins, et al.'s (1986) action theory incorporates goals, two processes for the stage of evaluation and two for the stage of execution.

Task Goal and Device Goal

The model assumes that skilled users have a schematic representation of the task that is in the form of a hierarchical structure involving two kinds of goals: *task goals* and *device goals*. Our goal representation is taken directly from the Yoked State Space Hypothesis proposed by Payne, Squibb, and Howes (1990). Payne, et al. assume that discovering how to carry out a task involves searching of two problem spaces. The first is a space of possible task states. The second is a space of possible device states that are required to achieve a given task state. We assume that each task goal is associated with one or more device goals. The device goals specify device states that must be achieved in order to satisfy an associated task goal.

Given a task goal and its associated device goals, the model simulates a sequence of action selections as follows.

Stage of Evaluation

Generating Display Representations

At first, the model generates a representation of the display. The display representation only includes information about identity of each object on the display and its appearance, e.g. highlighted, pointed-at, dragged, etc. *No* information about what actions can be taken on the object, or its meaning and relationships to other objects in the display is included in this initial display representation.

Elaborating the Display

All such information is generated by the elaboration process which retrieves information from long-term memory by *a random memory sampling process*. The retrieval cues are the representations of the current display, the task goal and the device goals. The probability that each cue retrieves particular information in a single memory retrieval process is proportional to the strength of the link between them. The model carries out multiple memory retrieval in a single elaboration process. A parameter, *the elaboration parameter*, controls the number of times each argument in the display and goal representations is used as retrieval cues[1].

The retrieved information elaborates the display representation, providing information about interrelationships between display objects, relationships between the task and display objects, and other attributes of display objects. *The elaborated display representation* is model's evaluation of the current display in the context defined by the task goal and the device goals.

Stage of Execution

Selecting Candidate Objects for Next Action

In the stage of execution, the model first limits its attention to a few number of screen objects out of ~100 objects displayed on the screen. These screen objects are candidates for the next action to be operated upon. The candidate object selection is performed on the basis of the evaluation, defined by the elaborated display representation. The model uses the spreading activation mechanism to select candidate objects. The process is dominated by two factors: the strengths of links from the representation of the goals, which is parametrized by a parameter, *the attention parameter*, and the number of propositions that are necessary to bridge the goals and the candidate objects[2].

---

[1]The model represents goals and display in propositions, like (is-on-screen OBJECT12). In the memory sampling process, the argument, such as OBJECT12, is used to retrieve information from long-term memory that has OBJECT12 as its argument.

[2]The model assumes the argument overlap mechanism to link up propositions. For example, the two propositions, (is-on-screen OBJECT12) and (has OBJECT12 CalculatorMenuItem), are linked by the shared argument, OBJECT12.

Selecting Action

The model considers all possible actions on each candidate object. The model incorporates 18 possible actions[3], such as "moving the mouse cursor to a menu item in order to display a pull-down menu." The process is dominated by the same two factors decribed above.

Furthermore, the action representations include conditions to be satisfied for their execution. The conditions are matched against the elaborated display representations. Some conditions are satisfied by the current screen, others by information that was retrieved from long-term memory in the elaboration process. For example, the model cannot select an action to double click a document icon for editing unless the icon is currently pointed at by the mouse cursor and the information is available that the icon can be double clicked. Observe that if information about a necessary condition is missing from an elaborated display representation, the model cannot perform that action on the *incorrectly* described object.

MODEL-BASED ANALYSIS OF ERRORS

In a set of experiments we conducted so far, where a graph drawing task was simulated, we found that the model could cause errors due to the following three reasons.

The first is that the process of selecting candidate objects for the next action fails to include the correct object on the list of candidate objects. The second possible cause of errors is that the correct action fails to become the highest activated action among executable actions. In the model's terms, these kinds of errors are ascribed to both or either of small values of the attention parameter (A), and /or missing bridging knowledge that had to be retrieved from long-term memory (B).

The third is that the elaboration process fails to incorporate all of the conditions for the correct action in the elaborated display representation. Low values of the elaboration parameter cause this error (C). Parameter values in the range of 12 to 20 caused the model to simulate error rates in the range of 10% to 20% (Kitajima and Polson, 1994, to appear). We argue that the elaboration parameter describes a speed-accuracy tradeoff process where low values of the parameter reduce the amount of time taken by the elaboration process.

COMPARISON WITH OTHER MODELS

The strength of the model is that the model generates correct actions as well as occasional errors without assuming a special set of mechanisms to produce erroneous actions. In this respect, the model is strikingly different from typical models of expert performance and error (Anderson, 1993; Reason, 1990; Card, et al., 1983). Typical models assume that skilled performance is mediated by detailed, large grain size action plans stored in long-term memory. Card, et al. (1983) refers to them as methods; Reason (1990) assumes that skilled performance is mediated by action schemata (Norman, 1981). Thus they have to be equipped with erroneous plans to generate errors. The grain size of action is much smaller in our model, at the level of individual pointing action. When the model makes an error, it has attempted to select a correct action based on incomplete knowledge, and/or insufficient attention. The incorrect action will be highly constrained by the user's current goals, the current state of the display, and the partial knowledge that was successfully retrieved from long-term memory. The candidate objects and the next action selected by a simulation are the model's best selections given the context represented by the elaborated display representation.

[3]Representations of actions define different functions of single physical actions in many different contexts. For simulating a graph drawing task, the model defines eighteen cognitive actions on six physical actions; Move-Mouse-Cursor, Single-Click, Double-Click, Hold-Mouse-Button-Down, Release-Mouse-Button, and Type.

MEASURING THE GULF OF EVALUATION IN DISPLAY-BASED HCI

In this section, we focus on the second cause of errror (B). We have experimental evidence that fits this issue. Experiments on experienced users learning a new application (Franzke, 1995) show that there is a subset of actions necessary to complete a well understood task that are difficult to discover by exploration. According to Hutchins, et al.'s theory of action, the GUI users must bridge the gulf of evaluation (Hutchins, Hollan, & Norman, 1986) by interpreting the current display in the context of their goals in order to perform a correct action on a correct object. In the language of Hutchins, et al.'s theory, first time users failed to build a bridge between their knowledge of the task and their current goal to the current state of the display and the actions available to them. On a second attempt at the task using the same interface, users with memory of successful interactions were able to do it smoothly (Franzke, 1995).

We argue that the degree of the gulf of evaluation should be reflected on the structure of the network representations used to select correct actions in our model. The necessary knowledge for correct actions, i.e. bridging knowledge and/or device goals, must have been acquired in the first attempt.

Simulation of Experienced Users

In the experimental sessions, reported in (Franzke, 1995), the subjects were told to create a line graph. After graph creation, they were shown four instruction cards each of which asked them to do a few modifications to the graph. Four applications were used for the experiments. The following sections describe the model's simulation of some of action selections necessary in the Excel 3.0 interaction.

Difficult Task – Creating a Line Graph

The following task included the most difficult steps.

TASK: create a line graph from the data in the spread-sheet.

Suppose that the subject has selected the spreadsheet cells containing the data for the plot. After this, the correct sequence of actions is (1) pointing at **File** in the menu bar and pulling down it; (2) pointing at **New...** menu item and releasing the mouse button. This action provides a dialog box with a selection list. Upon selection of **Chart**, another dialog box pops up which enables the subject to specify the data in the first column. This action sequence generates a default line graph which is then edited.

The model's simulation of (1) follows. To begin with, the model retrieves the device goals,

New... is displayed on the screen,
New... is selected.

The model can retrieve them because it has memory of successful experience. Notice that there are no routes from the task goal to the correct object since no labels (arguments) of the screen objects overlap with the task.

Then, the model retrieves representations of application specific knowledge from long-term memory. Among them,

File has an item **New...**,

is critical. The model can retrieve this because it has memory that relates the screen object **File** to **New...**.

The model can activate the correct object, **File**, by sending activation from the device goal to the retrieved knowledge, then to the correct object. Existence of both the device goal and the retrieved knowledge in the activation path is a necessary condition for the correct object to be selected.

Intermediate and Easy Task

Suppose that the subject is given an instruction for modification of the title of the graph.

TASK: change Letters in Title to Helvetica, 18.

A correct sequence of actions is (1) pointing at the title, **Data from XXX**, in the chart, currently displayed in Helvetica, 12, bold, and single-click it; (2) pointing at **Format** menu in the menu bar, and pull-down it; (3) dragging the mouse to **Font...**, and selecting it; (FONT dialog box pops up) (4) pointing at **18** in the scrolling list for size, and single-click it.

Intermediate Task – Selecting Title

The model retrieves knowledge,

Data from XXX is a kind of Title,

from long-term memory which is cued by the appearance of the title. This can be done if the subject has general knowledge about the structure of the chart, especially where title is supposed to appear in chart. This mediates the path to activate the correct object from the task goal.

Easy Task – Selecting an Item in List

In the interaction with the FONT dialog box, the argument in the task goal, 18, overlaps with the label of the correct object in the scrolling list. The correct display object has a direct activation path from the task goal.

Measuring The Gulf of Evaluation

The above analysis shows that the paths to activate the correct screen object depend on the relationship between the task goal and the object. It ranges from the closest situation where the task goal and the correct object share an argument, to the most remote situation where no direct path is available from the task goal to the correct object. In such cases, retrieval of device goals and application specific knowledge is needed to compensate the gap.

The current analysis suggests a way to measure the gulf of evaluation, defining it by the relatedness of the current task and the correct screen objects. It becomes smallest when the task goal is directly linked to the correct object. In such situation, both first time and experienced users have no difficulty (Franzke, 1995). On the other hand, the gulf becomes largest when both device goals and linking knowledge have to be retrieved from long-term memory. A large gap is impossible to discover for first time users, and even experienced users have a hard time to focus on the correct object (Franzke, 1995).

REFERENCES

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* NJ: Lawrence Erlbaum Associates.

Franzke, M. (1995) Turning Research into Practice: Characteristics of Display-Based Interaction. *In Proceedings of CHI '95*, ACM.

Hutchins, E.L., Hollan, J.D., & Norman, D.A. (1986). Direct manipulation interfaces. In Norman, D.A. & Draper, S.W., Eds. User Centered System Design. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kitajima, M. & Polson, P.G. (1992). A computational model of use of a graphical user interface. *In Proceedings of CHI '92*. ACM, pp. 241-249.

Kitajima, M. & Polson, P.G. (1994). A comprehension-based model of correct performance and errors in skilled, display-based human-computer interaction. *Technical Report # 94-02*, Institute of Cognitive Science, University of Colorado.

Kitajima, M. & Polson, P.G. (to appear). A comprehension-based model of correct performance and errors in skilled, display-based human-computer interaction. *IIJHCS.*

Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, **88**, 1-15.

Payne, S. J., Squibb, H. R., & Howes, A. (1990). The nature of device models: The yoked state hypothesis and some experiments with text editors. *Human-Computer Interaction,* **5**, 4, 415-444.

Reason, J. (1990). *Human Error.* New York, NY: Cambridge University Press.