Action Selection Error Probabilities in the Construction-Integration Theory-Based Model of Display-Based Human-Computer Interaction

Muneo Kitajima

National Institute of Bioscience and Human-Technology
1–1, Higashi, Tsukuba, Ibaraki, 305 JAPAN
kitajima@nibh.go.jp

ABSTRACT

The Kitajima and Polson model of display-based human-computer interaction (1992, 1994, in press) can make errors even if the model is provided with sufficient knowledge to perform correct actions. The major cause of errors is the failure to retrieve from long-term memory the knowledge necessary to select a correct action. This paper derives a set of mathematical formulae that calculate the probability of an error at a specific step in a task. This technique can replace a lengthy simulation runs that would be required to obtain reliable error probabilities. Each step in an example graph-drawing task is analyzed showing the factors that determine the likelihood of an error.

1. INTRODUCTION

Kitajima and Polson (1992, 1994, in press) have proposed a comprehension-based, performance model of skilled use of applications with graphical user interfaces like those of the Apple Macintosh and Microsoft Windows that accounts for both correct performance and errors made by expert users. The model is based on Hutchins, Hollan and Norman's (1986) action theory, in which the user's action selection process is modeled as a cyclic process, consisting of processes of (1) evaluating the current display by using the current goals and knowledge retrieved from long-term memory by a probabilistic memory sampling process, (2) selecting a small number of appropriate screen objects for a next action, and (3) selecting an appropriate action on one of the objects. The selected object and action are the outcome of a comprehension process applied to the current display. The model is an extension of the Mannes and Kintsch (1991) model of action planning which is based on the Kintsch's (1988) construction-integration theory of text comprehension.

In a set of simulation experiments reported in Kitajima and Polson (1992, 1994, in press), we found that the model makes errors even if it is provided with correct goals and sufficient information in long-term memory for selecting correct actions. Errors are caused by failures to retrieve critical information, such as affordances of screen objects, from long-term memory in the display evaluation process. This kind of error is called description error (Norman, 1981). The Kitajima and Polson (1992, 1994, in press) model simulates error rates of skilled users, in the range of 10% ~ 15% (Card, Moran and Newell, 1983).

In general, in order to obtain reliable error probabilities of action selections in a given *task context*, a large number of simulation runs would be required, where the task context is defined by the current goals, directly accessible knowledge in long-term memory, i.e. long-term working memory proposed by Ericsson and Kintsch (1995), and information displayed on the screen. However, we found that, if some conditions were satisfied, we could

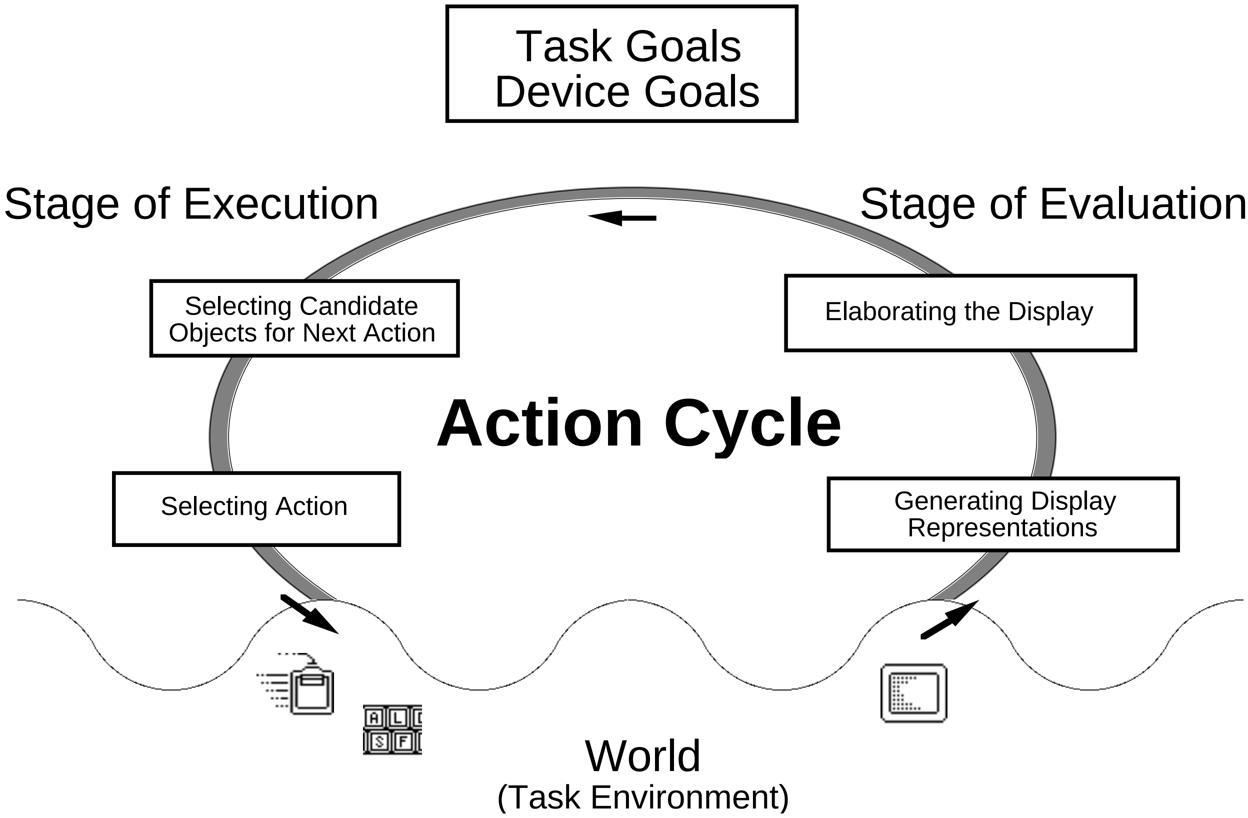

Figure 1. The comprehension-based, performance model of display-based HCI (Kitajima and Polson, in press), based on Hutchins, Hollan, & Norman's (1986) action cycle theory framework.

replace such a lengthy simulation runs with a single mathematical calculation because of the model's underlying mechanism of error generation.

The purpose of this paper is to provide a derivation of the mathematical formulae to calculate error probabilities of action selections for a given task context. The next section gives a brief introduction to the Kitajima and Polson's model of action selection in display-based HCI, followed by a section describing mathematical derivation of the formulae. Finally, a graph drawing task is analyzed showing how error probabilities vary with the interrelationships of the knowledge necessary for action selections.

2. THE MODEL OF DISPLAY-BASED HCI

Figure 1 illustrates the model. The model is based on Hutchins, Holland, and Norman's (1986) action theory framework, consisting of the following four basic components:

(1) *goals* representing what the user wants to accomplish, which are a schematic outline of the action sequence that will accomplish the task,

(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.

The Kitajima and Polson's model incorporates goals, two processes for the stage of evaluation and two for the stage of execution.

2.1 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 associated device goals specify device states that must be achieved in order to satisfy the task goal.

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

2.2 Stage of Evaluation

2.2.1 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 an object, or its meaning and relationships to other objects in the display is included in this initial display representation.

2.2.2 Elaborating the Display

All such information is generated by the elaboration process which retrieves information from long-term memory by *a random 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 performs multiple retrieval attempts during the elaboration process. A parameter, *the elaboration parameter*, $N_{elabotation}$, controls the number of times each element of the display and goal representations is used as retrieval cues[1].

The random sampling process is taken from Kintsch's (1988) construction-integration model. Kintsch used the Raaijmaker and Shiffrin's (1981) model to simulate the process of retrieving information from long-term memory for incorporation into the network. Kintsch assumed that the propositional representation of a sentence momentarily activates the meanings of words and other related information in long-term memory. This knowledge is incorporated in the network during the construction phase. Kintsch and Mross (1985) present evidence in favor of these assumptions.

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

2.3 Stage of Execution

In simulation of the stage of execution, the model selects three candidate screen objects and then selects an action-object pair. Each of these selection processes involves two phases: *construction* of a network of propositions, in which the network is linked up by *the argument overlap mechanism*[2], a principle from the Kintsch and van Dijk's (1978) model of text comprehension, followed by *integration* of the network in which an iterative, spreading activation mechanism is used to make the selection.

---

[1]The model represents goals and display in propositions, like OBJECT12 `is-on-screen`. 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]It is assumed that, when two propositions in the network share one argument, they are connected by a `link` of strength, the argument overlap weight. When they share *N* arguments, the strength is multiplied by *N*. For example, the two propositions, OBJECT12 `is-on-screen`, and OBJECT12 `has` CALCULATOR-MENU-ITEM, are linked by the shared argument, OBJECT12, and given one unit of the argument overlap weight for the link strength.

NAME:

`Point-at GRAPH in MENU-BAR with`
`ARROW-SHAPED-CURSOR`

CONDITION: IF

`GRAPH is-on-screen`
`GRAPH is-not-grabbed`
`GRAPH is-not TEXT-OBJECT` C-11
`POINTER-SHAPE is ARROW`
`POINTER-SHAPE is-not I-BEAM`

ACTION:

`GRAPH is-pointed-at`
`GRAPH is-on-screen`
`GRAPH is-not-grabbed`
`POINTER-SHAPE is ARROW`
`POINTER-SHAPE is-not I-BEAM`

Figure 2. Action representation for pointing at `Graph` menu. The condition, C-11, must be retrieved from long-term memory in order for this action to be executed.

2.3.1 Selecting Candidate Objects for Next Action

At first, the model limits its attention to a few number of screen objects out of ~100 objects displayed on the screen. The selected screen objects are candidates for the next action to be operated upon. The selection of candidate objects is performed in a network which is incorporating nodes representing the goals, the evaluation of the current display, and the candidate object nodes, one for each screen object.

The object selection process is dominated by two factors. The first factor is the strength of attention to the goals. The model assumes that strong attention characterizes expert behavior, that is, rapid and correct action selections (Kitajima and Polson, 1992, 1994). The model simulates the attention mechanism by a mechanism that reinforces the links between the goals and the other propositions. These special links are multiplied by *the attention parameter*. Large values of the attention parameter are required for the model to simulate expert behavior. The second is the number of propositions that are necessary to bridge the goals and the candidate objects. Note that the spreading activation mechanism can only activate significantly the nodes that are less than two links away from the goals[3]. The greater the attention parameter and the less the number of bridging propositions, the more activation flows to goal related propositions.

2.3.2 Selecting Action

The model considers all possible actions on each candidate object. The model incorporates 18 possible actions[4], such as "moving the mouse cursor to a menu item in order to display a pull-down menu." The selection of next action is performed in a network which is constructed by nodes representing the evaluation of the current display and all possible object-action pairs. The result of the activation process is dominated by the same two factors described above.

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 both of the following conditions are available in the elaborated display representations. The first is a screen condition, "the icon is currently pointed at by the mouse cursor", and the other is the information 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.

Figure 2 shows an example of action representations from our experiments, in which a graph drawing task using Cricket Graph, a Macintosh application, was simulated. The task description and a sequence of actions necessary for accomplishing the task are shown in the Appendix. Figure 2 illustrates the

---

[3] The nodes representing the current task goal, the current device goal, and the screen objects serve as activation sources. In the integration process, unit activation values allocated to the activation sources flow to the rest of the network. Due to a large value of the attention parameter, the goals become strong sources of activation.

[4] Representations of actions define different functions of single physical actions in many different contexts. For simulating the Cricket Graph 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.

action representation for pointing at `Graph` menu, which is the first step for the task. There are six conditions to be satisfied for its execution; the third one, GRAPH `is-not` TEXT-OBJECT, has to be retrieved from long-term memory, whereas the others are satisfied by the current screen. Action representations for eleven steps out of twelve for the Cricket Graph task, had only one condition to be retrieved from long-term memory, like this step.

3. ERROR PROBABILITY

3.1 Error Generation Mechanism

There are three ways the model can make errors. 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, because the node representing the correct object was not activated high enough to be selected as one of candidates. The second is that the correct action fails to become the highest activated action among executable actions through the spreading activation process. In the model's terms, these kinds of errors can be ascribed to missing bridging knowledge that had to be retrieved from long-term memory during the elaboration process.

The third is that the elaboration process fails to incorporate all of the necessary conditions for the correct action in the elaborated display representation. Low values of the elaboration parameter can cause this error. 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, in press). 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.

3.2 Mathematical Calculation of Error Probability

The probabilities of errors generated by the third cause can be calculated mathematically when a task context for action selections is specified. We derive the formulae starting with a detailed explanation of the mechanism of memory sampling process.

The retrieval process model was first described by Raaijmaker and Shiffrin (1981). In the elaboration process of our model, each element (argument) in each proposition, representing a goal or screen object, serves as a retrieval cue for propositions stored in long-term memory. Each argument in the cue propositions is used as the retrieval cue $N_{elaboration}$ times.

The retrieval probabilities are defined as follows. Let $C_n (1 \le n \le N)$ be the n-th condition to be retrieved from long-term memory for an action to be executable, where N is the number of such conditions. And let $X_{n,i} (1 \le i \le M_n)$ be the i-th retrieval cue for C_n. $X_{n,i}$, representing either the current task goal, the device goal, or one of the current screen objects, shares at least one argument with C_n. M_n is the number of such propositions from these retrieval cues. In the Cricket Graph task, M_n was around seven; N was three for the third step which is to select `Line` in the pull-down menu and one for the other eleven steps. The probability that proposition $X_{n,i}$ will retrieve proposition C_n, $P(C_n | X_{n,i})$, is given by the following formula;

$$P(C_n | X_{n,i}) = \frac{W_{X_{n,i},C_n}}{\sum_{X_j \in \{\text{all propositions but } X_{n,i}\}} W_{X_{n,i},X_j}}, \qquad (1)$$

where $W_{X_{n,i},X_j} \ge 0$ is the strength between proposition nodes $X_{n,i}$ and X_j, which is defined by the argument overlap weight, W, and the attention parameter, F.

Figure 3 illustrates interconnections among various components of the network. Each retrieval cue, $X_{n,i}$, is connected with propositions in long-term memory, and among themselves. The numbers in parentheses are typical link strengths used in the simulation experiments, assuming one overlapping argument between two interconnected propositions. As the result of very strong attention parameter values, $F = 16$, the goal related links, if they exist, have a tremendous effect on the results of memory sampling process. Namely, a display representation, if connected with a goal, cannot be a promising retrieval cue for a condition in long-term memory because most of its effort for memory sampling is absorbed by the strong link to the goal. In the simulation of the Cricket Graph task, $P(C_n | X_{n,i})$ ranged from 0.01 to 0.1.

	task-goal	device-goal	display	LTM
task-goal	(1)	$F^2 \times W$ (1024)	$F \times W$ (64)	$F \times W$ (64)
device-goal	$F^2 \times W$ (1024)	(1)	$F \times W$ (64)	$F \times W$ (64)
display	$F \times W$ (64)	$F \times W$ (64)	W (4)	W (4)
LTM	$F \times W$ (64)	$F \times W$ (64)	W (4)	W (4)

Figure 3. Interconnection among goals, display, and long-term memory. The numbers in parentheses are link strengths used in the simulation, being the attention parameter F is set to 16, and the argument overlap weight, W, set to 4, and assuming one overlapping argument between two components.

Next, we focus on an event, R_n, in which C_n is retrieved from long-term memory by one or more retrieval cues and in one or more retrieval attempts. Since these events are independent from each other, the probability of selecting a correct action by retrieving all the conditions is calculated by the following formula;

$$P(Correct\ Action) = 1 - P(Error) = P(R_1) \times \ldots \times P(R_n) \times \ldots \times P(R_N) \quad (2)$$

where $P(R_n)$ is the probability that the event R_n occurs.

Since the event R_n can happen when at least one of M_n cues successfully retrieves C_n, its probability is calculated as follows;

$$P(R_n) = 1 - \prod_{i=1}^{M_n} (1 - P_i(R_n)), \quad (3)$$

where $P_i(R_n)$ is the probability that i-th cue, $X_{n,i}$, causes the event R_n. Since $X_{n,i}$ is used as a retrieval cue $N_{elaboration}$ times and the probability that a single memory sampling trial successfully retrieves C_n is given by (1), $P_i(R_n)$ can be represented as follows;

$$P_i(R_n) = 1 - (1 - P(C_n | X_{n,i}))^{N_{elaboration}}. \quad (4)$$

Table 1. The retrieval cues for the first step of the Cricket Graph task and the number of links from each cue.

Cues in the Display Representations	number of links
`GRAPH is-on-screen`	12
`GRAPH isa SCREEN-OBJECT`	45
`GRAPH isa GRAPH-MENU-ITEM`	14
`GRAPH is-equal-to THE-GRAPH-TO-BE-POINTED-AT`	15
`GRAPH is-not-highlighted`	12
`GRAPH is-not-selected`	12
`GRAPH is-not-grabbed`	12

In summary, error probability for an action selection can be calculated by (2) when the strengths of links in the network for the task is specified. The probability distribution is defined by (1) when both the representations of the task goal, the device goals, the display, and long-term memory, and the value of attention parameter are specified. Note that the strength of argument overlap parameter, W, has no effect on the probability distribution.

4. CALCULATION OF ERROR PROBABILITIES FOR THE CRICKET GRAPH TASK

These formulae can be used to calculate error probabilities when we could expect errors only due to failure of retrieval of necessary conditions for the correct action. We found that this condition was satisfied in the simulation experiments we reported in detail in Kitajima and Polson (in press). They conducted 50 simulation runs for twelve steps of the Cricket Graph task with five elaboration parameter values; 3,000 action selections were performed in total. See the Appendix for the task description and the simulated sequence of actions. They obtained 396 errors, all of which except for one were due to missing conditions, namely, action slips.

Figure 4 shows error probabilities for the Cricket Graph task calculated by applying these formulae, which provides statistically more reliable error probabilities for the simulated situations than the experimentally obtained values from 3,000 simulation runs. By observing the figure, we can group the curves into four clusters. Each cluster is different from the others in terms of the degree of error

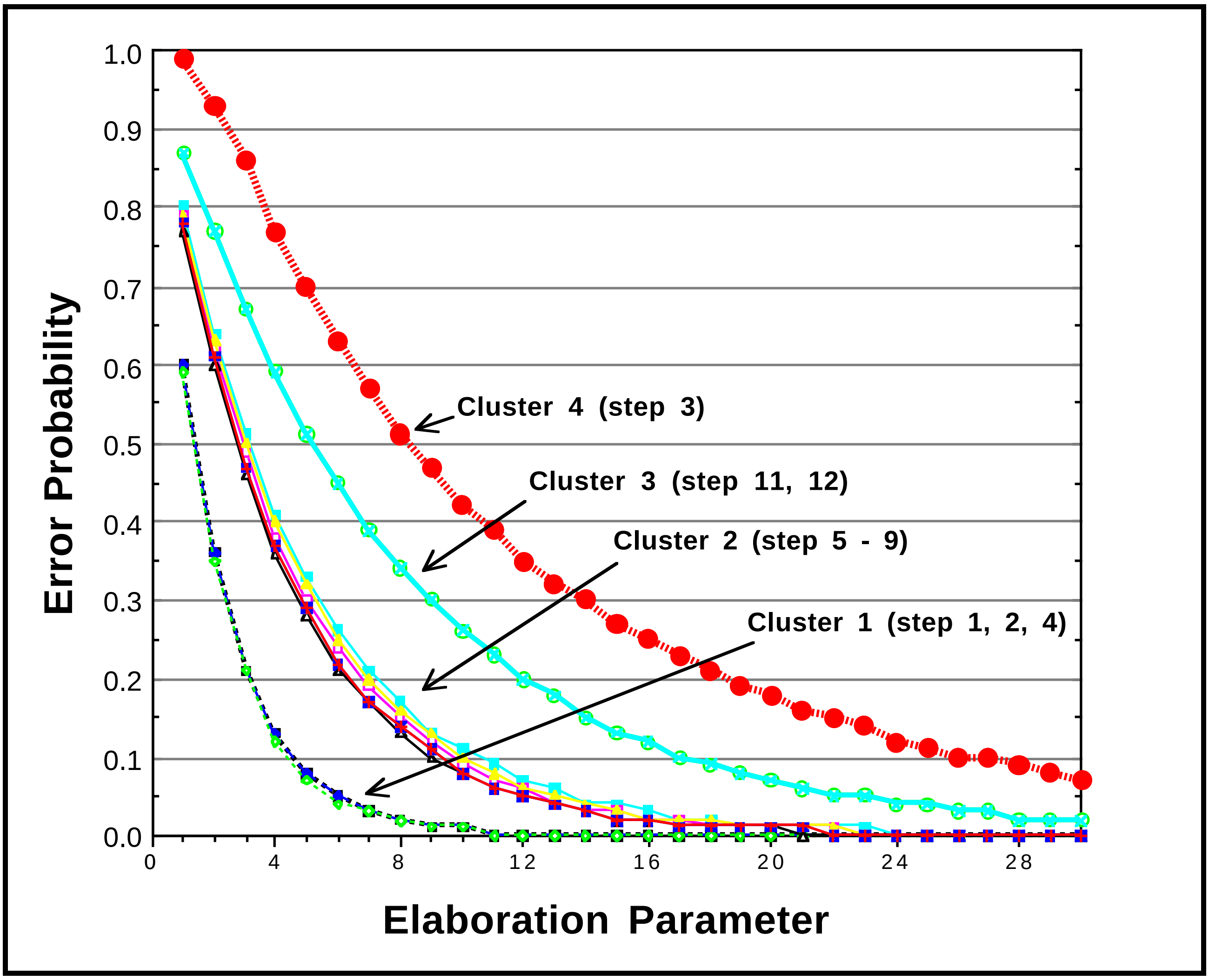

Figure 4. Error probabilities for each step in the Cricket Graph task as a function of the elaboration parameter.

proneness. In the following, the causes of differences between clusters are examined.

Cluster 1 is defined by action selections at steps 1, 2 and 4. Conditions for these actions consisted one proposition to be retrieved from long-term memory. None of the cues for the condition proposition was linked to a goal proposition. For example, step 1 requires the condition C-11 shown in Figure 2 for its execution which could be retrieved from seven cues in the display representations as shown in Table 1. Each cue was connected to propositions representing the display and those stored in long-term memory. The arguments, shown in small cap letters in Table 1, are used to establish linkages between them by the argument overlap mechanism. In Figure 3, two cells connecting display and, display or LTM, are relevant here. Since the model provides equal strength, W, to each link the retrieval probability of C-11 for each cue is calculated by inverting the number of links shown in the right column of Table 1 (see equation (1)). The probability that none of these cues retrieves the condition proposition for $N_{elaboration} = 1$ is given as follows;

$$P(Error(step1))$$
$$= 1 - P(R11; C-11 \text{ is retrieved from LTM})$$
$$= \left(1-\frac{1}{12}\right)^4 \times \left(1-\frac{1}{14}\right) \times \left(1-\frac{1}{15}\right) \times \left(1-\frac{1}{45}\right) = .5983$$

Cluster 2 includes action selections when the variables selection dialog box is on the screen (steps 5 through 10). Conditions for these actions required one proposition to be retrieved from long-term memory. There were seven cue propositions. One of them was the device goal, and the others six cues were from the display representation. The cues from the display representation had a single very strong link to the device goal which reduced the probability that each cue retrieves the condition proposition. For example, in step 5 (the action was "move pointer to `Serial Position` in the dialog box"), the retrieval probability for the condition proposition were 1/59 (the device goal), 1/26, 1/67, 1/27, 1/26, 1/26, and 1/29. The probability that none of these cues retrieves the condition proposition was .8004, which is the error probability for step 5 with $N_{elaboration} = 1$. Note that if connections to the device goal were not magnified by the attention parameter, the error probability becomes .6130, which is comparable to the one for the Cluster 1.

Cluster 3 contains steps 11 and 12 for selecting graph title and double click it. Conditions for these actions required one proposition to be retrieved from long-term memory. The number of cues was seven. Two of them were goals, and the others were from the display representation. The cues from the display representation had a double link to the task goal and the device goal. For step 11, the retrieval probability for the condition proposition were 1/64 (the task goal), 1/64 (the device goal), 1/39, 1/79, 1/74, 1/39, and 1/42. The probability that none of these cues retrieves the condition proposition was .8747 ($N_{elaboration} = 1$).

Cluster 4 is the step to select `Line` in the pull-down menu (step 3). There were three propositions to be retrieved from long-term memory, the first condition could be retrieved by nine cues, the second, by nine cues, and the third one, by one cue. All cues came from the display and none of them had links to either of goals. The probabilities of retrieval failure for each condition proposition were .5914, .5914, and .9167, respectively, and overall error probability became .9861 ($N_{elaboration} = 1$). Step 3 was difficult because of three conditions to be retrieved from long-term memory.

5. CONCLUSIONS

The following five factors determined the error probability for a given task context;

(1) the number of conditions to be retrieved from long-term memory,

(2) the number of retrieval cues,

(3) the presence/absence of links to goals from the cues in the display,

(4) the number of propositions that each cue is linked to (the fan effect),

(5) the number of retrieval attempts using a give cue.

Error probabilities for a particular step can be analyzed in terms of these factors. Figure 4 shows the effects of factors, (1), (3), and (5).

Predictions of error probabilities for a specific task context calculated by the formulae could be used to examine the plausibility of representations used for the task simulation. Remember that the representations have already been tested for sufficiency for producing correct actions. Even if a set of representations were proved to be sufficient for simulating correct task performance, it does not necessarily mean that it is *the correct set*. It might be close to the correct, but we may find implausible error probability predictions, which could then be used to improve the representations by considering the factors that cause the deviation.

For example, the current set of action representations lead to the prediction that step 3 was more difficult than steps 1, 2 and 4 because of the number of conditions to be retrieved from long-term memory was tripled. However, when user's action selections are observed in a laboratory study, it was found that this step is not difficult (Franzke, 1995). By combining knowledge from laboratory study concerning relative difficulty of different actions and the knowledge of the effect of the number of conditions to be retrieved on error probabilities, we will be able to improve action representations.

This analysis can also be used to examine the validity of the representations of goals and the display. The presence/absence of goal related links from the cues is critical for discriminating the degree of error proneness as shown in Figure 4. On the one hand, a large attention parameter is essential to activate selectively not only the correct actions but also the correct candidate objects. On the other hand, it reduces significantly the probabilities for retrieving condition proposition when its cues in the display are linked to the goals. This would be used to improve the representations of goals and display so that the model could simulate not only correct action selections but also correct error probabilities.

In conclusion, we derived a set of mathematical formulae to calculate error probabilities for a given task context, which was derived from the mechanism of error generation in Kitajima and Polson's (1994, in press) model. It provided a coherent way to re-examine the representations of task goals, device goals, display, long-term memory including actions, that have been passed the examination of sufficiency test for producing correct actions, from another measure of performance, i.e. error probabilities. We showed five factors that have effects on error probabilities, which provide appropriate directions for improving the representations that are valid not only for simulating correct actions but also erroneous actions.

ACKNOWLEDGMENT

The study reported here is part of collaborative research program involving the Institute of Cognitive Science, University of Colorado, and the author's organization, National Institute of Bioscience and Human-Technology, MITI, Japan. During initial phase of this collaboration, the author was a visitor at the Institute of Cognitive Science, University of Colorado, in 1991, supported by New Energy and Industrial Technology Development Organization.

REFERENCES

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

Ericsson, K. A. and Kintsch, W. (1995). Long-term working memory. *Psychological Review*, **102**, 211-245.

Franzke, M. (1995). Turning Research into Practice: Characteristics of Display-Based Interaction. *In Proceedings of CHI '95*, 421-428, 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*. 87-124. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Kintsch, W. & van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, **85**, 363-394.

Kintsch, W. & Mross, E. F. (1985). Context effects in word identification. *Journal of Memory and Language*, **24**, 336-349.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, **95**, 163-182.

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. *ICS Technical Report*. 94-02. Boulder, CO: Institute of Cognitive Science, University of Colorado.

Kitajima, M. & Polson, P. G. (in press). A comprehension-based model of correct performance and errors in skilled, display-based human-computer interaction. *International Journal of Human-Computer Systems*.

Mannes, S. M. & Kintsch, W. (1991). Routine computing tasks: Planning as understanding. *Cognitive Science*, **15**, 305-342.

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.

Raaijmaker, J. G. & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, **88**, 93-134.

[APPENDIX] CRICKET-GRAPH TASK

The major task studied in our simulation experiments involved preparing a graph that matches an example, shown in Figure A-1, using Cricket Graph 1.3. We briefly describe the task and summarize the model's representation of the action sequence necessary to accomplish it. We assume that the user is a skilled user of Cricket Graph and that he or she has been given the data to be plotted in a Cricket Graph document entitled "Example Data." Double-clicking "Example Data" causes the program to display a spreadsheet with three columns labeled "Observed," "Predicted," and "Serial Position." The user's task is to plot "Observed" as a function of "Serial Position" and then edit the resulting default graph so that it conforms to Figure A-1.

The user's first subtask, creating the default graph "Observed" plotted as a function of "Serial Position," involves selecting "Line-Graph" from the "Graph" pull-down menu which brings up a dialog box. The dialog box enables the user to designate the column labeled "Serial Position" as the X-axis and the column "Observed" as the Y-axis. Clicking a button labeled "New Plot" causes the default graph to be presented. The second major component of the task involves a sequence of editing operations that change X- and Y-axis ranges, the font and size of X- and Y-axis, legends, title, and the like. These editing operations enable the user to transform the default graph into a graph that matches the appearance of Figure A-1.

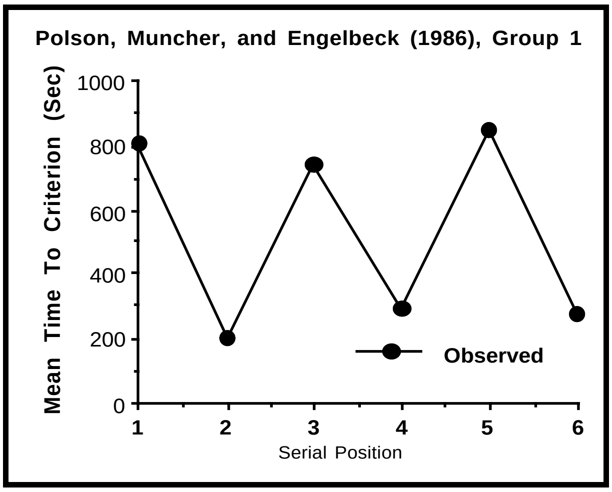

Figure A-1. The sample to be produced in the Cricket Graph task.

Table A-1 lists a correct sequence of actions with the representations of task goal and device goals.

TABLE A-1

Step No.	Task Goals (**TG**) and Device goals (**DG**)	Correct Action
	<u>Subtask 1</u> **TG-1**: to create a default line graph with Serial Position" as X axis versus "Observed" as Y axis	
1	**DG-11**: to see entry into the line graph environment	*Move Mouse Cursor* to `Graph`
2		*Press and Hold Mouse Button Down*
3		*Move Mouse Cursor* to `Line`
4		*Release Mouse Button*
5	**DG-12**: to see Serial Position is selected as X axis	*Move Mouse Cursor* to `Serial Position` in X axis selection list
6		*Single Click*
7	**DG-13**: to see Observed is selected as Y axis	*Move Mouse Cursor* to `Observed` in Y axis selection list
8		*Single Click*
9	**DG-14**: to see New-Plot is selected	*Move Mouse Cursor* to `New Plot`
10		*Single Click*
	<u>Subtask 2</u> **TG-2**: to edit the graph title	
11	**DG-21**: to see entry into the editing environment	*Move Mouse Cursor* to `Graph-Title`
12		*Double Click*