Applying Cognitive Psychology to Computer Systems
A Graduate Seminar in Psychology

Thomas P. Moran
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Stuart K. Card
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

This paper describes a graduate seminar given by the authors at Stanford University in the Department of Psychology in the Spring of 1980. Although the seminar was also listed as a Computer Science course and was attended by students and faculty from Computer Science and other departments, the course was addressed to Ph.D. students in cognitive psychology. The seminar assumed that the students were familiar with modern cognitive psychology, but did not assume that they necessarily had much experience with interactive computing.

Objectives. The focus of the seminar was on how the theories, methods, and results of cognitive psychology could be applied to a domain of practical interest—the design of interactive computing systems. Some specific objectives were that the students should:

—obtain some experience with interactive computing and get a feeling for the current state of user interface design;

—be able to analyze a user interface or a design problem to identify and formulate the psychological issues involved;

—understand the ways in which psychological knowledge can be brought to bear on user interface evaluation and design, especially the notion of calculational performance models;

—appreciate the issues in making psychological research both scientifically rigorous and relevant to design applications;

—understand current information-processing models of skilled user behavior;

—become familiar with the current research in human factors of computer systems.

Thus, the seminar was neither a survey nor a design course.

Format. The seminar, consisting of one three-hour class per week for ten weeks, was run in a workshop format. The assignment for each class was a set of readings (about three papers) and a problem set based on the readings. The problems consisted of analyses, small experiments, and calculations of various user interface performance issues. The class sessions mostly consisted of discussions of the problems (alternative approaches, critique of assumptions, comparison of results), but also included lectures on selected topics and seminar-style reports on the literature.

SESSION TOPICS

It is our view that an applied psychology of human-computer interaction should be based on task analysis and calculation from information-processing models [1]. Extending this view into the design of the course, we tried to cast the discussion topics into specific problems that would get the students actively involved in thinking about these issues. The emphasis on problems is apparent in the following summary of class sessions.

1. ***Introduction.*** Overview of issues and studies in human-computer interaction. Historical summary of early attempts to provide a foundation for the analysis and design of procedural skills, from Taylor and the Gilbreths through industrial engineering pre-determined time systems and current operator simulation systems.

2. ***Experience with Two Graphics Systems.*** In a four-hour evening laboratory session at Xerox PARC, students were given hands-on experience with two graphics systems, MARKUP and DRAW, which have different user interfaces. As they learned the systems and attempted to complete a set of tasks on them, students

kept records of their learning times and the cognitive difficulties they encountered. These systems provided a common ground of experiences for all members of the seminar to discuss and a domain for problems in the later sessions.

3. ***Psychological Issues in Human-Computer Interaction.*** Critique of the two drawing systems and listing of the psychological issues involved in the learning experience. The goal of this session was to make contact between cognitive psychology, and especially their particular interests in cognitive psychology, and human-computer interaction. The most important concept introduced here was that systems be evaluated by understanding their effects on user *performance.* Several dimensions of performance were discussed and applied to several common modes of human-computer dialogue.

4. ***Users' Conceptual Models.*** Nature of conceptual models, their psychological status, their representation, the role they play in user performance, how they can be evaluated, and how they can be designed. The problem was to sketch the basic conceptual models underlying the two drawing systems, using the framework of the Command Language Grammar [7], and to compare the two systems' models.

5. ***Learning Computer Systems.*** Factors affecting the learning of computer systems, including the rate at which learning proceeds. Consideration of what knowledge is acquired. The problems were: (5.1) Conduct a simple experiment by recalling the drawing system commands. (5.2) Estimate the number of tasks and methods learned in the drawing systems and compare the time-per-task learning rate with the results of Roberts [9]. (5.3) Test the power law of practice by performing and measuring a specific drawing task 20 times. (5.4) Analyze the consistency of the drawing systems by deriving a set of rules describing how to do a given set of tasks.

6. ***Models of Cognitive Skill.*** The nature of cognitive skill and how it relates to problems solving. The problems were: (6.1) Using a goal-operator (GOMS) analysis [2], calculate the time to do a given drawing task; propose a modification to the drawing system to improve performance on the task and calculate the improvement. (6.2) Compare the two drawing systems for how long they take to correct a certain common error. (6.3) Formulate rules for deciding between alternative methods for a given task. (6.4) Plot the user's working memory load during the execution of a given task. (6.5) Formulate the "problem space" for a subset of one of the drawing systems and explain what the user must know to be able to navigate through this space.

7. ***Practical Models of Performance Time.*** Computation of task time. Comparison of command language alternatives. Parametric and sensitivity analysis. The problems, using the Keystroke-Level Model [3], were: (7.1) Calculate the time to perform a given task; time how long the task actually takes; explain any discrepancy between the calculated and actual times. (7.2) Compute the efficiency of the drawing systems as the ratio of the calculated time for a given task and the minimum possible command specification. (7.3) Compute the ratio of the drawing task times between the two systems as a function of the number of line segments to be drawn. (7.4) Analyze a given segment of a detailed behavior protocol, and discuss how the behavior conforms to the Keystroke-Level Model.

8. ***Motor Skill Theory and Applications.*** Data entry and cursor moving techniques and devices. Theories of keying performance. Fitts's Law and theories of motor control. A guest lecture was given by Bill Verplank on manual control and linear systems theory. The problems were: (8.1) Validate Fitts's Law [4] for pointing time under several conditions. (8.2) Compute whether a system with a given processing rate is fast enough to track a mouse input device without hindering human performance. (8.3) Compute how much faster a touch typist will be on a given alphabetically arranged keyboard than on the QWERTY keyboard. (8.4) Compute how much improvement there will be on a given drawing task if a chordset is added to the terminal for specifying commands.

9. ***Error Analysis.*** Fitts and Jones's pilot error study [5]. Norman's theory of slips [8]. A guest lecture was given by Don Norman. The problems included the application of Norman's error taxonomy to the Fitts and Jones study and to errors in MARKUP and DRAW. Taking advantage of Norman's presence, the class also included a presentation and discussion of the Rumelhart and Norman [10] typing model.

10. ***Studies in Information Retrieval.*** Reviews by individual class members of the literature on psychological studies of information retrieval systems. Discussion included general issues, data organization, query languages, and natural language as a command language.

SAMPLE PROBLEMS

The problem sets were perhaps the most unique feature of this course. We tried to push the use of quantitative problems as far as we could, not only to give the students a way to think about the issues, but also to test notion of a calculation-based applied psychology. Below we give a sample of the problems given to the students (the problems are keyed to the session-problem numbers in the previous section):

Initial Learning of Procedures **(5.2).** This exercise attempts to use the results of Roberts to explain your initial learning time for MARKUP and DRAW. Consider only the subset of each system consisting of straight lines segments, end points on the grid, horizontal or vertical only, text labels, one font only, centered on grid points. Do the following for each system:

(a) From your learning log in Session 2, estimate the time it took you learn the above subset of the system.

(b) Make a list of the unit tasks that you learned and that are involved in the above subset of the system. How many distinct unit tasks are there?

(c) Take a sample of 5 to 10 unit tasks from (b) and write out the method (as a sequence of commands) for doing each task. Calculate the average number of commands per method, C.

(d) In Roberts [9] the learning rate is given as the time per unit task. Tentative results indicate that the best predictor of learning rate is the number of commands per task. Calculate the learning rate as $4.4C+1.8$ seconds. Now calculate the total learning time by multiplying by the total number of unit tasks from (b). How accurate is this calculation as a prediction of the actual learning time in (a)? Do you think that the prediction error is more in the rate calculation or in the estimated number of tasks learned?

(e) Be prepared to discuss: What factors are missing from this analysis of learning?

Consistency of Procedures **(5.4).** Users seem to be good at recognizing regularities or patterns in interacting with computer systems. This exercise is to explore the idea of user rules to describe interactions in a way that captures the consistencies that the user can learn.

(a) Write out the methods for doing each of the following tasks in both MARKUP and DRAW:

1. Create a line from P_1 to P_2 with thickness T.
2. Create a label at P with string S.
3. Change thickness of line L to T.
4. Delete line L.
5. Delete label L.
6. Erase the area within rectangle from P_1 to P_2.

The level chosen to describe the methods depends on the level at which consistencies can be found.

(b) Devise a set of rules for expressing the consistency among methods for each system. You can use linguistic context-free rewrite rules, BNF notation, CLG Interaction Level rules, or any other kind of production rule notation you prefer or which seems appropriate.

(c) What suggestions do you have for making MARKUP and DRAW more consistent?

(d) Be prepared to discuss: What kinds of learning mechanisms are needed to assimilate the rules of consistency?

Error Analysis **(6.2).** Consider the methods for the simple task of drawing a straight line in MARKUP and in DRAW.

(a) Do a Level 2 GOMS analysis of these methods for both MARKUP and DRAW. Calculate the time to do the task in each system. Which system is faster?

(b) Now consider that the user makes the error of drawing the line too long. Do a GOMS analysis of the correction of this error in both systems. Calculate the error time for each system. Which system is faster?

(c) In MARKUP, the user often makes the error of straying off the grid while drawing a line. From a GOMS analysis, calculate the error time to correct this kind of error. Combining this analysis with the one in (a), calculate the probability p of this kind of error in MARKUP that will make the task time in (a) the same for both systems.

Method Selection **(6.3).** Consider the task in DRAW of drawing n straight lines that are identical to n lines already on the display. There are at least two methods for doing this: (1) copy the existing lines or (2) draw them anew. Carry out a GOMS analysis of these alternative methods, and formulate selection rule(s) for choosing between the two methods.

Parametric Task Time Comparison **(7.3).** Consider the task of drawing a figure containing n vertical and horizontal line segments (vertical and horizontal segments alternating). What is the ratio between the time required to do the task in DRAW and the time required to do the task in MARKUP (a) as a function of n and (b) as a function of the average length of the segment L?

Fitts's Law **(8.1).** Suppose that a pointing device were to be designed which used only the finger muscles. What would you predict the net improvement in speed would be over the mouse? (Hint: Use Langolf's constant of 38 bits/sec for finger muscles.)

System Tracking Rate **(8.2).** A certain system must, for technical reasons, be designed such that the maximum rate at which it can track the mouse is 50 cm/sec. Is this rate fast enough so as not to interfere with human performance? (Hint: Use the Crossman and Goodeve derivation of Fitts's Law to compute the maximum movement velocity in pointing.)

COURSE MATERIALS

Three kinds of material were used in the course: the readings, the two graphics drawing systems, and the problems. We have already discussed the problems. The two drawing systems were important as a way of providing a common ground for discussion and analysis. Having two systems was useful, for the contrast between them highlighted issues and provided meaningful comparison problems. Also, the systems allowed the students to run experiments to check their calculations against actual behavior.

Most of the reading materials were not published when we ran the course, and there have been many new publications in the field since then. We will not list here what our current readings would be, except to mention a couple. The recent special issue of *Computing Surveys* [6] contains a representative overview of the current state of user psychology. The main source for our view of a theoretically-based, calculational user psychology is our forthcoming book [1].

REFERENCES

[1] Card, S. K.; Moran, T. P.; and Newell, A., (1982). *The Psychology of Human-Computer Interaction.* Hillsdale, N.J.: Erlbaum.

[2] Card, S. K.; Moran, T. P.; and Newell, A., (1980). Computer text-editing: an information-processing analysis of a routine cognitive skill. *Cognitive Psychology 12*, 396-410.

[3] Card, S. K.; Moran, T. P.; and Newell, A., (1980). The Keystroke-Level Model for user performance time with interactive systems. *Communications of the ACM 23*, 396-410.

[4] Card, S.; English, W. K.; Burr, B. J., (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics 21,* 601-613.

[5] Fitts, P. M., and Jones, R. E., (1947). Analysis of factors contributing to 460 "pilot-error" experiences in operating aircraft controls. Reprinted in *Selected papers on human factors in the design and use of control systems*, H. W. Sinaiko, editor. New York: Dover, 1961.

[6] Moran, T. P., editor, (1981). *The Psychology of the Computer User.* Special Issue of *Computing Surveys*, vol. 13, no. 1, (available from ACM).

[7] Moran, T. P., (1981). The Command Language Grammar: a representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies 15*, 3-50.

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

[9] Roberts, T. R., (1979). Evaluation of computer text editors. Ph.D. dissertation, Stanford University. Also, Report SSL-79-9, Xerox PARC.

[10] Rumelhart, D. E., and Norman, D. A., (1981). Simulating a skilled typist: a study of skilled cognitive-motor performance. CHIP Report 102, Department of Psychology, University of California, San Diego.