Modeling the Effect of Task and Graphical Representation on Response Latency in a Graph Reading Task

David Peebles, University of Huddersfield, Queensgate, Huddersfield, United Kingdom, and **Peter C.-H. Cheng,** University of Sussex, Brighton, United Kingdom

We report an investigation into the processes involved in a common graph-reading task using two types of Cartesian graph. We describe an experiment and eye movement study, the results of which show that optimal scan paths assumed in the task analysis approximate the detailed sequences of saccades made by individuals. The research demonstrates the computational inequivalence of two sets of informationally equivalent graphs and illustrates how the computational advantages of a representation outweigh factors such as user unfamiliarity. We describe two models, using the ACT rational perceptual motor (ACT-R/PM) cognitive architecture, that replicate the pattern of observed response latencies and the complex scan paths revealed by the eye movement study. Finally, we outline three guidelines for designers of visual displays: Designers should (a) consider how different quantities are encoded within any chosen representational format, (b) consider the full range of alternative varieties of a given task, and (c) balance the cost of familiarization with the computational advantages of less familiar representations. Actual or potential applications of this research include informing the design and selection of appropriate visual displays and illustrating the practice and utility of task analysis, eye tracking, and cognitive modeling for understanding interactive tasks with external representations.

INTRODUCTION

The ability to interpret and reason with charts and graphs is increasingly important in this information-rich society. Charts, graphs, and other diagrams are used extensively in science, engineering, business, and the media. To be able to reason with charts and graphs effectively requires sophisticated perceptual and reasoning skills and a broad range of general and specific knowledge about diagrammatic representations. Understanding diagrammatic reasoning is an important goal for cognitive scientists, therefore, not only because of the ubiquity of diagrammatic representations but also because diagrammatic reasoning is a process in which behavior is a function of a complex interaction among three factors: the cognitive and perceptual skills of the reasoner, the graphical properties of the external representation being used, and the specific requirements of the task being undertaken. It is also likely that a deeper understanding of the relationships among these three factors will have implications for the design of more effective graphical representations.

In the area of graph-based reasoning, we have carried out a number of investigations into how these three factors affect reasoning with different types of Cartesian coordinate (*x,y*) graphs (Peebles & Cheng, 2001, 2002; Peebles, Cheng, & Shadbolt, 1999) and have proposed the graph-based reasoning (GBR) model to characterize the complex interactions and resulting behavior. The goals of GBR are similar to those of the

Address correspondence to David Peebles, Department of Behavioural Sciences, University of Huddersfield, Queensgate, Huddersfield, HD1 3DH, UK; D.Peebles@hud.ac.uk. **HUMAN FACTORS,** Vol. 45, No. 1, Spring 2003, pp. 28–46.

cognition-artifact-task (Gray & Altmann, 2001) and embodied cognition-artifact-task (Byrne, 2001; Gray, 2000; Gray & Boehm-Davis, 2000) frameworks proposed to characterize interactive behavior in human-computer interaction tasks.

Figure 1 shows the four graphs used in the experiment reported here. They depict the amount (in millions of units) of UK offshore oil and gas production over two decades. In the function graphs the argument variable (AV: time in years) is represented on the *x* axis and the quantity variables (QV: oil and gas) on the *y* axis, whereas in the parametric graphs the quantity variables are represented on the *x* and *y* axes and time is plotted as a parameterizing variable along the curve.

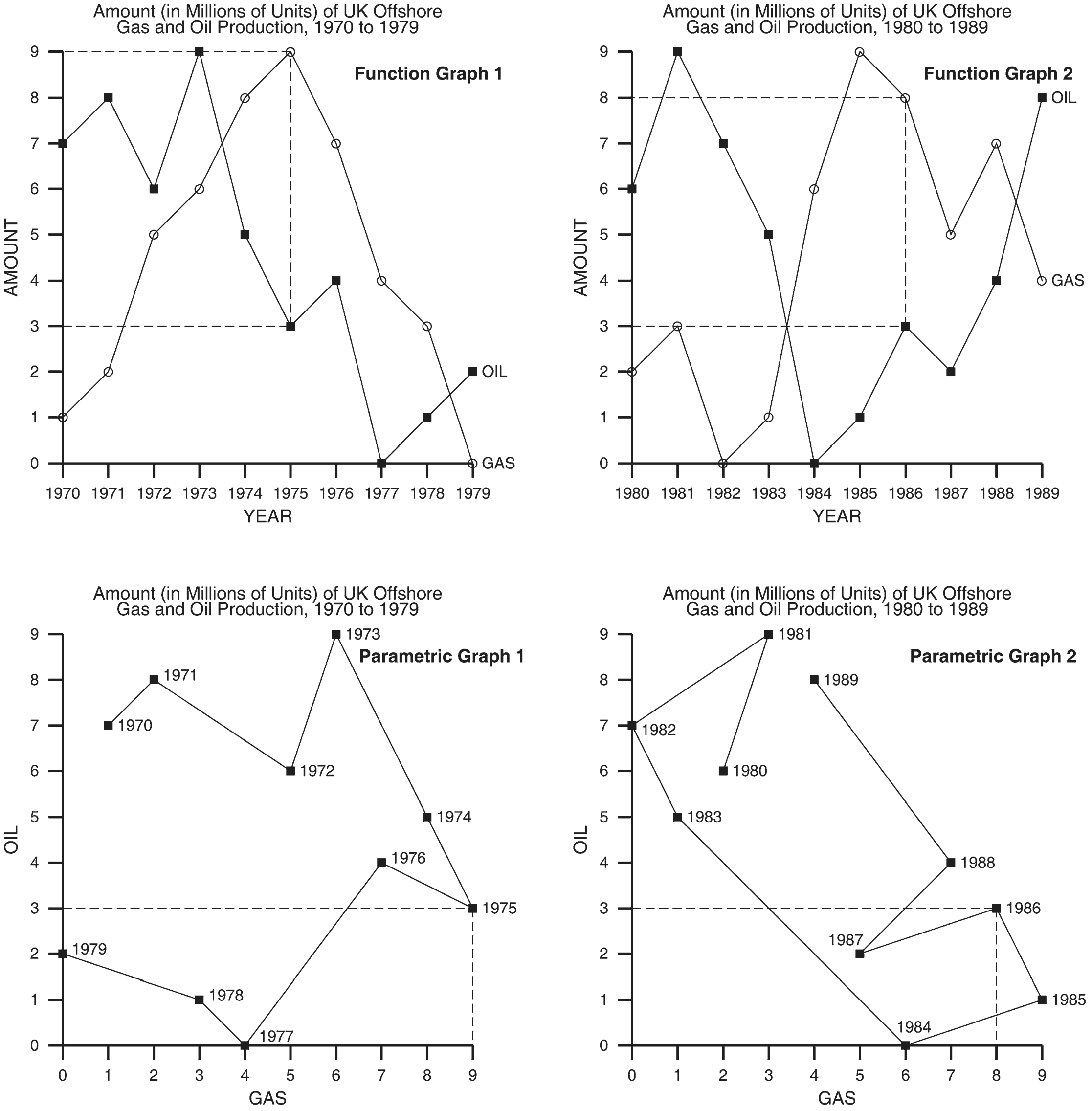

Figure 1. Function and parametric graphs used in the experiment depicting values of oil and gas (the quantity variables, QVs) for each year (the argument variable, AV). The graphs on the left (labeled 1) show years 1970 to 1979, and those on the right (labeled 2), 1980 to 1989. Dashed lines indicate the optimal scan path required to answer the question, "When the value of oil is 3, what is the value of gas?"

To evaluate the similarity between function and parametric graphs, we start with the notions of informational and computational equivalence of representations defined by Larkin and Simon (1987). According to their definition, two representations are *informationally equivalent* if no information can be inferred from one that cannot be inferred from the other and if each can be constructed from the information in the other. Conversely, two representations are *computationally equivalent* if they are informationally equivalent and if any inferences that can be drawn "quickly and easily" from the explicit information in one can be similarly drawn from the explicit information in the other, and vice versa. This latter term requires that not only the information content of the representations be taken into account but also the nature and speed of the various operators used to interact with them. According to Larkin and Simon's criteria, each pair of function and parametric graphs generated from the same data set for this study can be considered to be informationally equivalent. The computational equivalence of the graphs, however, is one of the issues this research seeks to address.

Function and parametric graphs also share several important properties. They are both simple line graphs using a two-dimensional Cartesian coordinate system to relate quantities and represent magnitudes. Although the two graph types assign different variables to their axes, they are similar in that both represent specific values of plotted variables as points on the line. Because of these visual and representational similarities, many of the basic operators for accessing items of information are the same for both graph types. It is also quite likely that these similarities invoke similar general graph schemas and interpretative processes (Kosslyn, 1989; Pinker, 1990) and that inferences from both graph types are influenced by the same set of biases (Carpenter & Shah, 1998; Gattis & Holyoak, 1996; Shah & Carpenter, 1995).

In previous experiments, however, we have tested a wide range of tasks with function and parametric graphs and demonstrated that despite their similarities, consistent and substantial differences do occur in task completion times, error rates, and patterns of error (Peebles & Cheng, 2001, 2002; Peebles et al., 1999). The GBR model explains why many of these differences occur in terms of the visual scan paths that users may follow through the graph when carrying out a task. GBR assumes that given a particular information retrieval task, experienced graph users will retrieve a procedure for obtaining that information involving a sequence of saccades and fixations to the target location. The resulting scan path will be more or less optimal depending on the user's general graph knowledge, the user's familiarity with the particular graph type being used, or the concepts and procedures required by the task. For example, the scan path produced by a user less familiar with graphs in general or with a particular graph type can be expected to be suboptimal in the sense that it would be less direct, indicating that a procedure involving random search or a step-by-step testing of individual points on the graph was being employed.

The GBR model's analysis of the graph-based reasoning tasks we studied involves the production of a cognitive task analysis specifying the procedural steps required to perform the task and the scan path that would occur should these steps be carried out. For each task, the scan path consists of a sequence of eye movements that minimize the number of saccades and fixations to obtain the solution or reach the target location in the graph. This approach is similar to that adopted in Lohse's (1993) understanding cognitive information engineering (UCIE) model of graphical perception. In more recent experiments it has also been possible to predict incorrect task solutions by generating the scan paths that occur when common procedural errors are made. Using these optimal and erroneous scan paths, GBR can then be used to predict which of the graph types should facilitate fewer errors or more rapid responses for a particular task.

To illustrate, consider the task of retrieving the value of gas when the value of oil is 3 using the function and parametric graphs in Figure 1. With the function graphs, once the given value of oil has been located on the *y* axis, three saccades (indicated by the dashed lines in Figure 1) are required to (a) locate the associated point on the oil line, (b) identify the corresponding point on the gas line at the same *x* coordinate, and (c) identify the required value of gas on the *y* axis. With the parametric graphs, however, once the given value of oil has been located on the *y* axis, the process requires only two saccades, one to locate the point on the year line at $y = 3$ and one to identify the corresponding value of gas on the *x* axis. According to this analysis, therefore, users of parametric graphs should be more rapid and accurate than function-graph users because of the fewer saccades required by the parametric graph and the greater number of possible incorrect saccades that function-graph users may make. These predictions have been borne out consistently in our experiments.

The analyses provided by GBR are useful in delineating the structure of tasks and generating general explanations for the relative differences in errors and completion times for the same task using different graph types. As it stands, however, the approach has a number of limitations. First, it remains an open question whether the assumption of an optimal scan path glosses over important cognitive and strategic factors at an individual level. For example, graph users may be required to re-encode items of information that have been lost from working memory during the course of processing. In addition, given that graph users are aware that information is available for rescanning at all times, it is possible that they may make a strategic decision to trade off additional saccades for a reduction in working memory load. If this is the case, then the current analyses may miss an important level of detail that would shed light on the cognitive load that these tasks are imposing and the strategies by which graph users optimize their retrieval procedures.

A second limitation of the approach is that GBR's temporal predictions remain at the level of statements that a task will, on average, take longer to complete when using one graph than when using another. A more powerful model would produce quantitative predictions of task completion times. There are at least two ways to do this. The first involves obtaining previously documented times for the various steps in the task-analytic model to generate a predicted total response time (RT) for each task as the sum of all the times for each of the component steps in one's model. This is the approach employed in several of the goals, operators, methods, and selection rules (GOMS) class of task analysis techniques (Card, Moran, & Newell, 1983; John & Kieras, 1994; Olson & Olson, 1990) and, in the area of graph-based reasoning, that adopted by Lohse (1993) in his UCIE model. By adding cognitive and timing parameters to a GOMS analysis of graphical perception, Lohse produced a model that simulated certain question-answering procedures using line graphs, bar graphs, and tables and then predicted question-answering times by assuming an optimal sequence of eye movements to scan the graphical representation.

This approach has proved useful in predicting execution times for a variety of tasks, but it has a number of constraints (John & Kieras, 1994). First, it assumes that reliable latency estimates are available for all the component tasks in the model. Second, it assumes that the users being modeled are well practiced and make no errors during the task. These constraints can be difficult to satisfy when modeling novel tasks, tasks involving components without prior estimates, or users unfamiliar with a particular graph type, for example.

An alternative method of predicting RTs for graph-based reasoning tasks is to construct cognitive process models using a programmable cognitive architecture that has the ability to represent the environment and capture the complex interactions among the graphical object, cognition, and perceptual-motor behavior. Several such computational theories of how these interactions are controlled – termed theories of *embodied cognition* (Kieras & Meyer, 1997) – have recently been developed: executive-process/interactive control (or EPIC; see Kieras & Meyer, 1997), EPIC-Soar (Chong & Laird, 1997), and ACT rational perceptual motor (ACT-R/PM; (Byrne & Anderson, 1998).

Constructing computational models that are grounded in cognitive theory enables one to incorporate and test relevant cognitive factors (e.g., the required declarative and procedural knowledge, the strategies adopted, and working memory limitations) as well as perceptual-motor factors, such as mouse movements and shifts in visual attention. Like other cognitive task-analytic approaches, computational theories incorporate assumptions about the execution latencies for component unit tasks and so can make precise predictions about the total time to complete individual tasks. Unlike other approaches, however, computational models are able to execute the task, providing an important sufficiency proof that the model accounts for the task. In addition, a number of architectures contain learning mechanisms, enabling them to model various effects of practice on performance.

In this article we present the results of a graph-based reasoning experiment and cognitive process models of the experiment in an attempt to address these limitations. In the experiment participants were asked to perform simple tasks using function and parametric graphs that would be predicted, based on the optimality assumptions currently employed in GBR, to produce varying response patterns by requiring different fixation sequences. To address the issue of whether these optimality assumptions are justified, some participants' eye movements were recorded as they solved the problems. We show that although the RT and error data are in line with GBR's predictions, certain patterns in the eye movement data do not follow the optimal sequences assumed in the current GBR model.

We then describe two computational models of the experiment, created using the ACT-R/PM cognitive architecture, that account for the observed patterns of eye fixations and that produce similar patterns of RTs for the various experimental conditions. The models are based on the GBR task analysis and incorporate a number of assumptions about learning the locations of graph elements, learning the associations among them, the limitations of working memory, and strategic choices. Also, in matching the observed RT data, we show how these assumptions determine which aspects of the graphical representation are encoded during the course of the experiment and which, through either memory limitations or strategic choice, are not. Finally, the models are used to provide detailed explanations of the RT differences between experimental conditions.

EXPERIMENT

A common task when using a graph is to elicit the value of one variable corresponding to a given value of another. This task was chosen for the experiment because it is a basic graph-reading skill that users learn and because the procedures involved are relatively simple. The knowledge required to carry out these tasks concerns primarily the sequence of fixations required to reach the *given location* in the graph, representing the given value of the given variable, and then to reach the *target location,* representing the corresponding value of the required variable. In previous research, however, we have discovered that the effectiveness of a particular graphical representation for retrieving the required information depends on the details of the task (i.e., which variable is given and which is sought; Peebles et al., 1999).

Method

Design. The experiment was a mixed design with one between-subjects variable (graph type: function or parametric) and two within-subjects variables: question type (three levels corresponding to the three possible combinations of given and required variables) and graph type (two levels representing different decades – the 1970s and 1980s, represented by the graphs on the left and right of Figure 1, respectively). Participants were randomly allocated to one of the two graph type conditions.

Participants. The participants were 44 undergraduate and postgraduate psychology students from the University of Nottingham who were paid £3 to take part in the experiment; an additional 4 participants were paid £5 to take part in the eye movement study.

Materials. The experiment was carried out using two PC computers with 17-inch (43-cm) displays and an SMI iView eye tracker using a RED II desktop pupil/corneal reflectance tracker (SensoMotoric Instruments GmbH, Teltow/Berlin, Germany) with a sampling rate of 50 Hz. This system records eye movements at 20-ms intervals remotely from a position in front of the computer display. The system has an automatic head movement compensation mechanism, but we also restrained participants' heads in a frame fixed to the table in order to reduce recording error that would be caused by head movement.

The stimuli used in the experiment were the four graphs shown in Figure 1. The graphs and data sets were designed so that the argument variable (AV, year) and the two quantity variables (QV, oil and gas) all had 10 values ranging from 0 to 9, and the full range of these values was represented by the data points for oil and gas in both decades. A set of questions was produced using all the values for the three variables in each decade. The questions had the same basic structure, giving one variable's value and requiring the value of another corresponding variable. All permutations of variables were used, producing a total of 120 questions (3 given variables × 2 required variables × 10 variable values × 2 graphs). For the analysis these questions were assigned one of three question type codes: QV-QV, QV-AV, and AV-QV, according to which variable's value was given and which was required, respectively.

Participants were seated approximately 80 cm from the 72-ppi computer display. The graphs were 15.5 cm square (including axis labels), corresponding to approximately 11.1° of visual angle. The characters representing variable values were 0.4 cm high (approximately 0.21° of visual angle), and those for the axis labels and questions were 0.4 and 0.5 cm high (approximately 0.29° and 0.36° of visual angle), respectively. Axis ticks were spaced 1.5 cm (approximately 1.1° of visual angle) apart.

Procedure. During the experiment the two graphs in a condition were presented alternately (the first graph being selected at random) so that participants saw each graph on every other trial. On each trial a graph was presented with a question above it. The questions were presented in a form requiring a minimum amount of text. For example, the QV-QV question "gas = 2, oil = ?" required the value of oil when gas is equal to 2 to be found, whereas the AV-QV question "year= 1978, gas = ?" asked the participant to find the value of gas in 1978.

Participants were required to answer the same 60 questions twice, once for each decade. The order of these 120 questions was randomized across the set of trials. In QV-AV questions, which required a year value, the final digit of the required value was represented by a question mark (e.g., "gas = 6, year = 197?" or "oil = 3, year = 198?"), and participants were instructed to enter only the final digit of the target year. Each question element was centered on a coordinate point that remained invariant throughout the experiment; approximately 3.5 cm (about 2.5 of visual angle) separated the centers of adjacent text items.

With the graph and question, a button labeled "answer" appeared in the top right corner of the window. Participants were instructed to click on this answer button as soon as they had obtained the answer to the question. RTs were recorded from the onset of a question to the mouse click on the answer button. When this button was clicked, the button, graph, and question were removed from the screen and a circle of buttons labeled clockwise from 0 to 9 appeared, centered on the answer button. Participants entered their answers by clicking the appropriate number button. When the number button was clicked, the next graph, question, and answer button appeared on the screen. This method was devised so that participants in the eye movement study would not have to take their eyes away from the screen to enter answers, as would be the case if they had used the keyboard.

Before starting the experiment participants were directed to answer the questions as rapidly and as accurately as possible, and they were given time to become familiar with the graphs and to practice entering numbers using the circle of number buttons and mouse.

Results

Participants' graph familiarity. Immediately after taking part in the experiment, participants were asked to provide two ratings of their familiarity with the graph type they had been using. They were required to rate the frequency with which they normally encountered information presented in the form just seen (i.e., how often they had come across this graph type) by choosing high, medium, or low frequency. They were also asked to rate, on a scale from 1 to 9, how familiar they considered themselves to be with the type of graph they had just encountered (1 = *very unfamiliar,* 9 = *very familiar*). The graph exposure frequency and mean familiarity ratings for the two conditions are displayed in Table 1.

An analysis of variance (ANOVA) on these data revealed that the ratings of graph familiarity for the function condition participants were

TABLE 1: Participants' Graph Encounter Frequency and Mean (and *SD*) Familiarity Ratings for the Two Graph Conditions

Rating	Function	Parametric
Low	9	12
Medium	8	9
High	5	1
Familiarity	6.73 (1.58)	4.50 (2.28)

significantly higher than were those of the parametric condition participants, $F(1, 42) = 14.16$, $p < .001$. In line with this, the encounter frequency ratings showed that more of the function condition participants rated their encounters with the graph type as high and fewer rated them as low.

If a new or infrequently encountered graph is observed – even if it shares many properties with other, more familiar graphs – it is likely that time must be taken to interpret the particular representational features of the graph in addition to the time taken to interpret the relationships being represented. If this is the case, then a further indirect measure of the participants' relative familiarity with function and parametric graph types would be the average amount of time they took to familiarize themselves with the two graphs they were using. Before starting the trials, participants were presented with the two graphs in random order and instructed to look at each graph until they were satisfied that they understood it and what it represented, at which point they pressed a key on the keyboard to erase the graph and proceed. Figure 2 shows the mean time taken by participants in each condition to become familiar with the graphs as a function of presentation order.

Figure 2 reveals that familiarization time for the second graph fell significantly, by about 10 s, in both conditions, $F(1, 80) = 14.51$, $p < .001$, and that the parametric condition participants required more time than did function condition participants to familiarize themselves with both the first and second graph (on average, 2.4 and 4.2 s, respectively). Although these differences are not statistically significant, the trend of these differences is consistent with participants' familiarity and exposure frequency ratings at the end

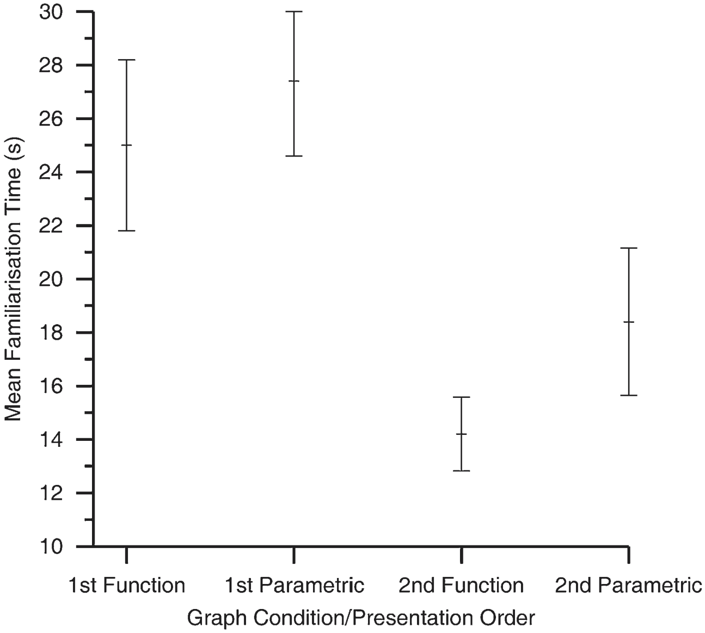

Figure 2. Mean time taken by participants in each graph condition to become familiar with experimental graphs as a function of presentation order. Error bars indicate standard error. Note that the *y* axis values do not start at 0.

of the experiment. These results indicate that although undergraduate psychology students are relatively less familiar with parametric-type graphs than with function graphs, the amount of time they take to understand the less familiar graphs to their satisfaction is not significantly increased. These factors will be considered when interpreting the correct response (CR) and RT data.

Response accuracy and latency data. The proportions of CRs and mean RTs for each question type for the two graphs in each condition are presented in Table 2. Although the data reveal high levels of accuracy for all three question types in both graph conditions, an ANOVA on the response accuracy data revealed a significant effect of question type, $F(2, 239) = 28.187$, $p < .01$, indicating that some question types were generally more demanding than others. In both graph conditions, participants made more errors with the QV-QV task than with the other two and responded most accurately to the AV-QV task.

Although there is little variability in the accuracy of responses between conditions, the time participants took to respond varies significantly, both between graph conditions and within each condition, according to the type of question being attempted. An ANOVA on the RT data revealed significant main effects of question type, $F(2, 239) = 18.447$, $p < .01$, and graph number, $F(1, 239) = 5.76$, $p < .05$, and significant interactions between graph type and question type, $F(2, 239) = 36.314$, $p < .01$, and among graph type, question type, and graph number, $F(2, 239) = 3.913$, $p < .05$.

Although the main effect of question type can be accounted for in terms of the number of procedural steps for each question defined by the task analysis, explanations for the effect of graph number are more speculative at the moment. The data show that responses using Graph 2 were generally slower than those using Graph 1, particularly in the function condition. This may be the result of differences in the perceptual features of the graphs produced by the two data sets.

For example, in Graph 1, the two plotted lines are relatively differentiable, as the line representing gas forms a distinct “hill” shape, rising to a peak and falling again. It may be the case that the plot lines in Graph 2 are less differentiable, as they are more similar, neither having such a distinct shape. In the parametric conditions it is only the RTs for the AV-QV questions that are substantially slower for Graph 2. Because in this case the given value is the year, this slowing may be attributable to the fact that the sequence of years is counterclockwise along the curve (see Figure 1, Parametric Graph 2). Currently these accounts are only reasonable hypotheses and can be tested by a detailed analysis of the eye movement data at a later date.

The general pattern of errors and RTs is consistent with the optimal scan path assumption in the current GBR analysis. As this analysis also forms the basis for the ACT-R/PM models, further explication is required. For this analysis and the following analysis of the eye movement data, it is necessary to divide the experimental display into seven regions. The regions, shown in Figure 3, were the same for

TABLE 2: Mean (*SD*) Correct Responses (CR) and Response Times (RT) for Each Graph Condition and Question Type

	Mean CR				Mean RT (s)			
Question Type	Function		Parametric		Function		Parametric	
	Graph 1	Graph 2	Graph 1	Graph 2	Graph 1	Graph 2	Graph 1	Graph 2
QV-QV	.911	.898	.925	.891	5.93	6.18	4.89	4.83
	(.285)	(.303)	(.264)	(.312)	(2.33)	(2.11)	(2.00)	(1.97)
AV-QV	.970	.982	.993	.989	5.01	5.07	4.88	5.17
	(.170)	(.134)	(.082)	(.106)	(1.78)	(1.64)	(1.57)	(1.65)
QV-AV	.943	.930	.939	.925	5.51	5.80	4.38	4.41
	(.232)	(.256)	(.240)	(.264)	(1.89)	(2.77)	(1.71)	(1.54)

Note: Graph 1 and Graph 2 refer to the graphs as illustrated in Figure 1, not to presentation order.

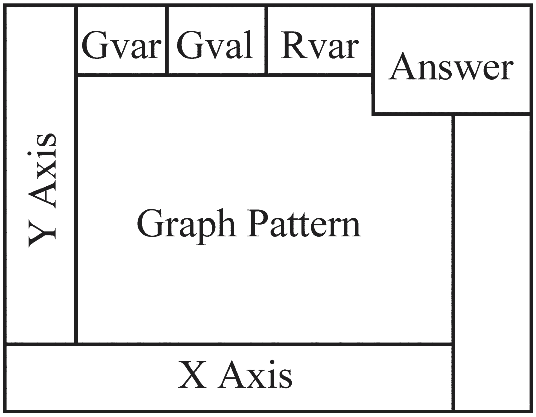

Figure 3. Seven regions of the experimental display defined for the fixation analysis (Gvar = given variable, Gval = given value, Rvar = required variable).

all four graphs and define the relevant units of the display: *x* axis, *y* axis, graph pattern, answer buttons, and three regions representing the three elements of the question – given variable, given value, and required variable. For several analyses the three question element regions were combined to produce a single question region. This approach is similar to that employed by Carpenter and Shah (1998) in their investigation of graph comprehension.

An individual trial in the experiment can be divided into six basic subgoals: (a) read the question; (b) identify the *start location,* determined by the given variable; (c) identify the *given location,* representing the given value of the given variable; (d) from the given location, identify the *target location,* representing the required variable; (e) identify the *target value* at the target location; and (f) enter the answer. For all the question types in the function graphs and the QV-QV and QV-AV questions in the parametric graphs, the start location is an axis and the given location is a point on this axis representing the given value. For AV-QV questions in the parametric graphs, however, the start location is the plot region of the graph and the given location is a point on the plotted line. Once at the given location, the participant must find the corresponding target location, and it is at this stage in the process that the main differences between the question and graph types exist and, consequently, where variations in the RTs arise.

For example, when answering QV-QV questions using a function graph, as illustrated in Figure 1, in order to reach the target location one must locate the correct plot point for the given value, find the associated plot point for the required variable, and then return to the *y* axis to identify the value of this point. To answer the same questions using the parametric graphs, however, one is required only to identify the correct plot point for the given value and then find the value of this point on the alternative axis. Consequently, the smaller mean RT for QV-QV questions in the parametric condition can be accounted for in terms of the shorter scan path for parametric graphs (question, *x/y* axis, graph, *x/y* axis, answer) as compared with that for function graphs (question, *y* axis, graph, graph, *y* axis, answer).

A similar explanation may be provided for the significantly greater time required to answer QV-AV questions using the function graphs. With function graphs, to get to the target location one must identify the correct plot point for the given value and then find the value of this point on the *x* axis. Conversely, with parametric graphs the target values are within the same fixation region as the plot point of the given value, thereby reducing the number of cognitive and perceptual steps required to fixate on the target location. In this case the optimal sequence of fixations is predicted to be question, *x/y* axis, graph, answer; whereas that for the function graphs is question, *y* axis, graph, *x* axis, answer.

Finally, for the AV-QV questions, the relative rapidity with which function graph users are able to answer these questions, as compared with the other question types, is attributable to the fact that they are able to rapidly identify the given year on the *x* axis and then carry out the two-step process of identifying the target point on the correct line and retrieving its value from the *y* axis. The optimal sequence of fixations for this procedure is question, *x* axis, graph, *y* axis, answer. The data show that this procedure takes approximately the same time as the corresponding procedure for the parametric graphs, which require search for the given year in the graph and then retrieval of its value from the target axis, the optimal fixation sequence of this procedure being question, graph, *x/y* axis, answer.

The experiment demonstrates that despite the numerous similarities that exist between function and parametric graphs, the type of graph used can significantly affect the time it takes to retrieve the required information and that this effect is dependent on the nature of the task. It is also evident that the probability of retrieving incorrect information depends on specific details of the task – that is, which variable is given in the question and which variable value is being sought. The GBR model is able to explain these differences in terms of a detailed task analysis using the assumption of an optimal scan path through the graph to the target location. The eye movement study is designed to determine whether this assumption is justified by analyzing the actual scan paths produced when the tasks are performed.

Eye movement data. The eye movement data were analyzed by computing the frequency and duration of fixations in each of the seven regions defined over the graphical display and recording the pattern of transitions among these regions during individual trials. For the analysis we adopt Carpenter and Shah's (1998) term *gaze* to refer to a sequence of consecutive fixations on a display region that is unbroken by fixations in other regions. The raw *x,y* coordinate data were aggregated into gazes with a minimum duration of 100 ms, a value large enough to eliminate most saccades, short fixations, and noise and yet still capture all relevant fixations. Fixations of 100 ms or more in each region were recorded, and a scan path consisting of the sequence of gazes for each trial was produced. From a total of 480 trials in the eye movement study, 28 were removed because the analysis produced an unusable scan path.

Several interesting patterns emerge from the analysis of these gaze sequences. First, the average number of transitions between regions for all question types, shown in Table 3, is consistently greater than the optimal number predicted by GBR. For all question types, and irrespective of the graph type being used, participants made an average of three to four additional transitions to reach the solution. In the majority of cases these additional transitions were between either the axes and the graph or the question and the graph; participants rarely fixated on the answer region until entering an answer. In 31% of all trials, participants made at least one additional gaze on an axis after having previously fixated on that axis and then the graph. More detailed examination revealed that in the majority of cases, participants had fixated on an axis value, proceeded to the plot point corresponding to that value, and then made an additional saccade back to the axis to check that the value was in line with the point.

A second interesting pattern involves the encoding of the question elements. In 62.7% of all trials, and irrespective of the graph used and question type being attempted, participants made at least one additional gaze on the question after having initially gazed on the question and subsequently the graph. Of these, 48.0% involved one further transition, 11.9% involved two additional transitions, and 2.8% involved three or more transitions.

Examination of the gazes on individual question elements shows that these patterns are the result of a combination of strategies adopted by the participants. For all questions, participants scanned the question elements from right to left as their point of regard moved from its current position at the answer buttons from the previous trial. The eye movement data show that

TABLE 3: Mean Number of Gaze Transitions Between Display Regions for Function and Parametric Graphs Observed for Each Question Type, and the Optimal Number Predicted by the GBR Model

	Function		Parametric	
Question Type	Observed	Optimal	Observed	Optimal
QV-QV	7.66	5.0	8.21	5.0
AV-QV	7.86	5.0	8.90	4.0
QV-AV	8.05	5.0	8.05	4.0

participants either read the required variable, the given value, and then the given variable or scanned past the required variable to read the given value first and then the given variable, reading the required variable only later in the problem-solving process, when the given location had been reached. This suggests two explanations for the re-encoding of question elements.

In the first case, participants initially encode the three elements of the question but are required to re-encode certain parts of it that cannot be retrieved from working memory because of the cognitive load involved in carrying out the problem-solving procedures (see Peebles & Cheng, 2001, for a more detailed discussion). In the second case, participants effectively break the problem into two sections, the first to get to the given location in the graph and the second to move from the given location to the target location corresponding to the solution. It is also possible that the observed gaze patterns may result from a combination of these factors if, during the course of the experiment, participants adopt the latter strategy in order to minimize the number of question element retrieval failures.

The eye movement data also show that if the given variable is a year value, participants typically do not then scan to the given variable location (as it is always a year), whereas if the given value is not a year value, then they always scan to the given variable to determine whether it is oil or gas.

ACT-R/PM MODELS OF THE EXPERIMENT

One of the main aims of this research is to construct models of graph-based reasoning that are grounded in cognitive theory and incorporate cognitive factors such as memory decay and interference together with perceptual-motor components that provide realistic interactive behavior. ACT-R/PM has the required cognitive and perceptual mechanisms with which to develop such models. The goal is to use the task analysis provided by GBR to provide the basic set of constraints for construction of the ACT-R cognitive model, a methodology that has been previously employed to produce models of several tasks (see Anderson, 1993). We then use the eye movement and response time data from the experiment to provide an empirical test of the cognitive model as well as an additional source of hypotheses concerning the strategic and cognitive factors assumed in the model. In this way we go beyond the optimal analysis that has been assumed in the current GBR model to produce a detailed and testable account of the eye movements, gaze durations, and task completion times.

ACT-R/PM (Byrne & Anderson, 1998), an extension of the ACT-R cognitive architecture (Anderson, 1993; Anderson & Lebiere, 1998), adds perceptual-motor modules to the central ACT-R cognitive module. Three modules, based on the corresponding modules of EPIC (Kieras & Meyer, 1997), provide ACT-R with rudimentary speech and audition capabilities as well as elements of motor control to simulate manual interaction with a computer keyboard and mouse. ACT-R/PM's visual module is the most developed perceptual component. An extension of the original ACT-R visual interface, the visual module allows the modeling of visual attention shifts to objects on a computer display. In a recent development these perceptual-motor modules have been fully integrated into the ACT-R 5.0 architecture. This is the version of the architecture used for the models presented here.

Space limitations permit only a brief outline of the most relevant aspects of ACT-R/PM here, but detailed discussions can be found in Anderson and Lebiere (1998) and Byrne (2001). ACT-R contains two memory systems: a procedural memory consisting of a set of *productions* and a declarative memory in the form of a network of *chunks.* The system also contains five *buffers* that store information about such things as the current goal, the item of declarative knowledge that is currently available to the system, and the current state of the perceptual and motor modules. Each buffer may contain only one item of information, as each new request for new information replaces the current contents of the buffer. Productions are rules of the form "IF <condition> THEN <action>," the condition specifying chunks that must be present for the rule to apply and the action specifying the actions to be taken should this occur. The conditions of productions are typically tests of the contents of the various buffers, whereas on the action side these contents can be modified, the current goal terminated and a new goal set, or a request made for the retrieval of a chunk from declarative memory.

ACT-R/PM combines serial and parallel processing, the cognitive and perceptual-motor modules of ACT-R/PM being for the most part serial, with the modules running in parallel with each other. The various processes also have associated latency parameters. For example, the default time of a production to fire is 50 ms, whereas the time taken to move the mouse cursor to an object on the computer screen is calculated using Fitts' law (Fitts, 1954). ACT-R/PM's visual modules represent the display image (which is constructed in the LISP programming language) as a visual icon, and productions are able to direct visual attention to elements of this icon. When attention is focused on an object in the icon, declarative chunks representing that object and its location are created and placed in the system's visual and visual location buffers, respectively. While they are the current contents of buffers, these chunks can be accessed by productions. Each chunk is created with an initial activation value so that when a chunk is not in a buffer, as long as this value is above a certain threshold value, it can be retrieved by the cognitive system and placed in a buffer.

ACT-R's declarative memory has an activation-based retrieval process and includes a mechanism by which the activation of chunks decreases over time. However, the activation of a visual object or visual location chunk is increased when visual attention is refocused on the visual object that it represents. ACT-R has been used to model a wide range of cognitive phenomena (Anderson & Lebiere, 1998), and in recent years, with the inclusion of the perceptual-motor modules, it has been applied to a number of complex interactive tasks in the area of human-computer interaction and human factors research (e.g., Byrne, 2001; Savucci, 2001b; Schoelles & Gray, 2000).

A Description of the Models

ACT-R/PM models of the experiment's two graph conditions were constructed that were able to interact with an exact replica of the software used to run the experiment. The models consist of two sets of productions to carry out the six subgoals outlined earlier: one set to carry out the four subgoals related to answering the questions and another set of general productions to read the question at the start of a trial and enter the answer when the required value has been obtained. The main diagrammatic operators embodied in the productions are the same for both models (e.g., searching for specific axis labels and values, reading and comparing graphical and textual elements with the contents of various buffers, scanning from axis values to plotted points and vice versa). The primary difference between the two models is the control structure that sequences the execution of these operators. The productions for reading the question and entering the answer are shared by both models.

The processing of individual questions involves three main operations: (a) a series of memory retrieval requests for the elements of the question and the location of these elements in the graph, (b) search procedures, and (c) visual attention processes directed by the current goal of the system. As these sequences determine the output of the models for each graph and question condition, we now describe in detail the six subgoals involved in the process.

Read the question. Both models contain the same set of productions to read the question elements from right to left at the start of each trial. The number of question elements read by a model can vary. A model may opt to read the required variable at the start of a trial or scan straight to the given value. In addition, if the given value is a year, the model will not read the given variable.

Identify the start location. When the question elements have been read, a new goal is set to identify the starting location determined by the given variable. If the system can remember which axis represents it, a retrieval request is made to recall the given value and a new goal is set to identify its location. If the axis cannot be remembered, however, then the model must search the graph to find the appropriate axis label. This also occurs in the parametric model when the given value is a year. If the system is unable to retrieve the year's location in the graph from declarative memory, a systematic search for the year is initiated in the plot region of the graph until it is found. This leads to some important effects on the RTs for the AV-QV questions, which will be discussed further.

Identify the given location. Before the goal to identify the given location can be carried out, the given value must be remembered. Because a certain amount of time has elapsed since the given value was read, the activation of the chunk – and consequently the probability that it can be retrieved – will be less (Peebles & Cheng, 2001). If it cannot be retrieved from declarative memory, the question element is reread and the focus of attention returned to its previous location. With the chunk now sufficiently active, the original goal can be pursued and a search for the given axis value carried out. Once the given location is identified, a new goal is set to identify the target location.

Identify the target location. As with previous goals, the appropriate question element (in this case the required variable) must be in the system's buffer before the target location can be identified. If the strategy adopted by the model is to skip reading the required variable until this stage of the process, the question element is read for the first time. If, however, the required variable was read at the start of the trial, because even more time has elapsed there will be a higher probability that its chunk cannot be retrieved and must be re-encoded.

Once the chunk is in the buffer, what happens next depends on the graph and question type. In the function graph condition the system obtains the symbol associated with a QV and then directs visual attention to the appropriate symbol having the same coordinate value as the current location on the current axis. In the QV-QV questions, an additional step is required whereby visual attention is then directed to the alternative symbol having the same coordinate value on the x axis as the current symbol. In the parametric graph condition, given that there is only one kind of plot symbol, for the QV-QV and QV-AV questions visual attention is simply directed to the symbol with the same coordinate value as the current axis location. For AV-QV questions, however, visual attention is at a year value in the graph, and so attention is directed to the plot point nearest to the current location.

Identify the target value and enter the answer. For QV-AV questions in the parametric condition, the target value is the year label nearest the currently attended symbol. For all other question and graph conditions, when the target location has been reached, the target value is an axis tick label at the same x or y coordinate as the currently attended symbol.

When the target value has been identified, visual attention and the mouse cursor are moved to the answer button and the button is clicked. As in the experiment, task completion time is recorded at this point. When the response buttons appear, the correct one is identified and clicked, at which time the next trial starts.

How Learning Affects the Models' Behavior over Time

Although we have not focused on the issue of learning here, it is clear that the mechanisms provided in ACT-R allow for learning to take place and that this may be used to account for the characteristic decrease in task completion time found over the course of the experiments.

In particular, the learning of symbol-variable associations and the locations of graph elements, such as the axis representing individual variables and the locations of axis values, are captured by the mechanisms of the models. For example, during the early stages of the experiment, when the models do not know which axis represents which variable, they are required to scan the axis labels. Later, when the axis label chunks have been strengthened sufficiently by repeated visiting, retrieval requests are frequently successful and the models are able to remember consistently the correct axis for a given variable.

A similar situation occurs in the function graph condition concerning the identification of the plot symbol representing each of the QVs. During each trial a retrieval request is made for the plot symbol that represents a QV. If it cannot be retrieved, a goal is set to look for a plot symbol label in the graph region, and when one is found, the associative chunks representing each variable-symbol pair are strengthened. Over time these chunks are sufficiently active to be recalled consistently.

In relation to the locations of axis values, the models are provided with prior knowledge of the general region where values are located on x and y axes. This knowledge guides the search for a particular value, but if another value is found, its location is stored and the search reinitiated in the same region, providing an element of stochasticity to the search procedure. Eventually, however, during the course of the experiment the precise location of each value on the *x* and *y* axes can be retrieved, and the search time is thus reduced.

This learning mechanism is a significant factor in the RT predictions produced by the models. For example, the learning of question element locations is crucial to the parametric graph model's account of the increased RT for the AV-QV questions, as compared with RTs for the other questions. For the QV-QV and QV-AV questions, by the last half of the trials, the axis representing each QV has been learned, as has the location of the individual values on the axis. Although the location chunk of each year value is strengthened each time it is attended to, the frequency of these strengthenings is not sufficient to keep them active enough to be recalled. Hence in the majority of cases for AV-QV questions, a memory retrieval request for the year value fails and a search must be made, increasing the time to complete the task. It is possible that this may result from a strategic decision not to attempt to retrieve the location of a year, but the resulting behavior would be virtually identical.

Comparing the Models with the Experimental Data

From the foregoing discussion, it is clear that the behavior of the models is a result of a number of assumptions regarding the complex interplay of memory retrieval requests, search procedures, and visual attention processes. As with all cognitive models, the ACT-R/PM models of these tasks require, and allow, the specification of these component cognitive and perceptual-motor steps at a much finer grain size than that required by GBR. The behavior of the ACT-R/PM models is also dependent on assumptions about how much of the question is remembered during a trial and how much of the diagram is remembered during the course of the experiment. Whether these assumptions fit with the experimental data is an important question, which we now address.

Modeling task completion times. One of our primary aims in constructing these cognitive models is to move beyond the qualitative predictions of relative task completion times produced by GBR to make precise and testable quantitative RT predictions for different types of graph and task. Our initial concern when evaluating the models, therefore, is to determine whether the RT predictions of the models are comparable to those of relatively well practiced individuals who are familiar with both the task and their respective graphs. To this end, the mean RT for the second half of the experiment (a total of 60 trials) was computed for each question condition. To produce a data set that could be compared with that from the experimental participants, the function and parametric models were both run 20 times and the mean RT for the last 60 trials was computed. These mean observed and model RTs for function and parametric graphs are presented in Figure 4.

When running the models we set the various free parameters to their default values, except for the *base-level learning* parameter, which controls the rate of learning and decay for the base-level activation of declarative chunks. This parameter affects the rate of forgetting of the question and graph elements and, consequently, the number of re-encodings that are required during a trial. The default value of this parameter (.5) resulted in the models' learning too slowly, increasing the number of re-encodings and consequently the overall task completion times. To improve the fit of the mean RTs to those from the experimental data, the value of this parameter was gradually incremented until the value of .93 was found to increase the learning rate and reduce the number of re-encodings sufficiently to provide an acceptable fit.

In general, the RTs from the models for individual question conditions are close to those from the experiment, and the overall pattern of relative task completion times resembles that of the observed data, although the differences between the question conditions for the function graphs are smaller for the model than for the observed differences. Both parametric and function graph models provide reasonable fits to the data, accounting for 87% (R^2 = .868, *RMSE* = .123) and 66% (R^2 = .664, *RMSE* = .199) of the

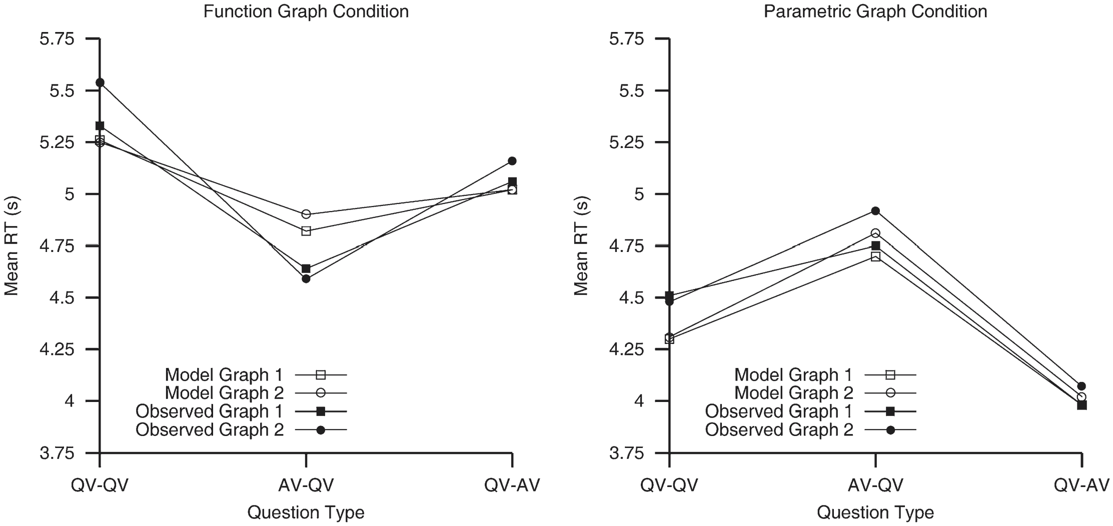

Figure 4. Response times for experimental participants and ACT-R/PM models for each question type averaged over the last 60 trials of the experiment. Note that the *y* axis values do not start at 0.

variance in the observed RT data, respectively. The condition in which the data from the model diverge most from the observed data is for AV-QV questions in the function graph condition. Although the model does predict that the mean RTs for these graphs are the smallest in this condition – because (a) the number of initial question elements that need to be read is fewer than for the other conditions, and (b) at the stage when the target location is identified, a memory retrieval request is not required, whereas it is for the other question types – the predicted RT is greater than that of the observed data.

Explaining eye movement data. The eye movement data reveal two principal reasons for the encoding of question elements during the course of a trial. The first involves a strategic choice to delay encoding the required variable until it is required, and the second involves encoding the three question elements at the start of a trial and the subsequent rereading of individual elements during the trial. The ACT-R/PM models are able to capture both these processes by different means. The first is modeled by productions that encode each question element at the appropriate stage. The second process is captured by the decay of base-level activation of perceptual chunks during the time course of problem solving.

On these trials the models initially encode all three question elements. At each stage of the problem, however, at least one element must be retrieved from memory in order for the model to proceed. From the time of their initial encoding, the question element chunks' activation is decaying, and consequently the probability that they cannot be retrieved increases. As the trial progresses the probability that a question element will not be retrieved and must be re-encoded increases. Because the required variable is most often the one retrieved near the end of a trial, this may account for the 48% of trials requiring a further gaze to the question region and the adoption of the strategy delaying the required variable encoding. A more detailed account of this process can be found in Peebles and Cheng (2001, 2002).

To illustrate this point, Figures 5 and 6 compare screen shots of model scan paths and eye movements recorded from one participant for two questions using the 1980s parametric graph. The participant screen shots are taken from a viewer application in which the eye movement data are displayed over a bitmap of the display where all the answer buttons are displayed simultaneously and the locations of the question elements are represented by variables (*QM* stands for question mark). Each arrow on the eye movement scan path represents a data point

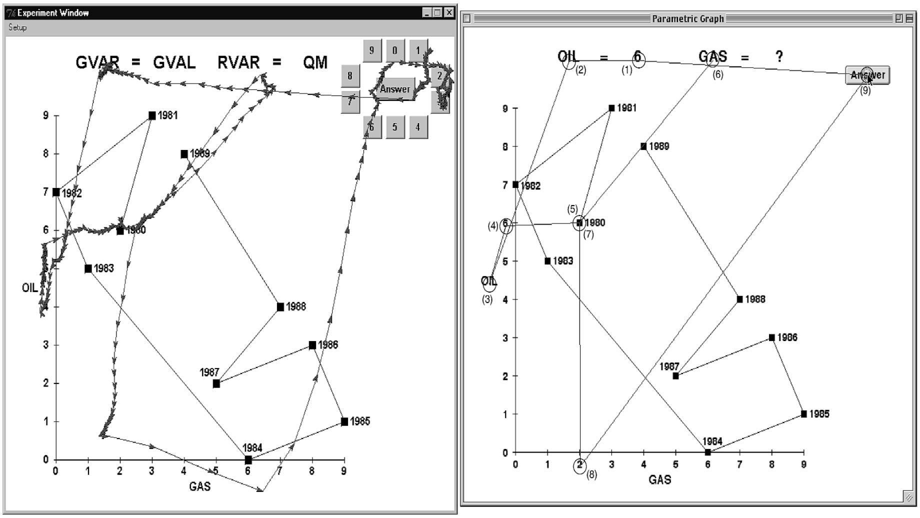

Figure 5. Screen shots showing an experimental participant's eye movement data (left) and the ACT-R/PM model's visual attention scan path (right) for the QV-QV question "oil = 6, gas = ?" using the 1980s parametric graph. In the model screen shot, numbered circles on the scan path indicate the location and sequence of fixations.

recorded at 20-ms intervals. The model screen shot is taken immediately after visual attention has shifted to the answer button once the required value has been found. The numbered circles on the scan path indicate the sequence of fixations produced by the model.

The two model screen shots illustrate the situation described earlier in which the required value is read near the end of the trial. In both examples the model starts from the answer region, looks directly to the given value (Fixation 1) and then the given variable (Fixation 2), checks

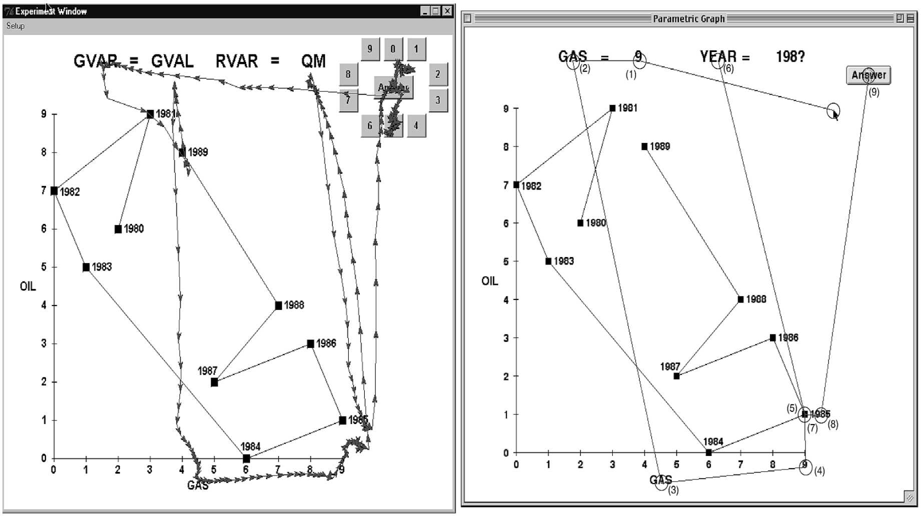

Figure 6. Screen shots showing an experimental participant's eye movement data (left) and the ACT-R/PM model's visual attention scan path (right) for the QV-AV question "gas = 9, year = 198?" using the 1980s parametric graph. In the model screen shot, numbered circles on the scan path indicate the location and sequence of fixations.

that the given variable is represented on the axis (Fixation 3), and then proceeds to the given value (Fixation 4) and the given location (Fixation 5). As the model now needs to know what the required variable is, it looks for it (Fixation 6), returns to the given location (Fixation 7), and then proceeds to identify the associated required value (Fixation 8) and the answer button (Fixation 9). In both cases the pattern of eye movements produced by the participant is strikingly similar to that of the model.

DISCUSSION

The research reported here has several important implications for the design of visual displays and graphical user interfaces. It also provides a clear illustration of the value of eye movement data and computational modeling techniques in the analysis of complex interactive behavior. We now discuss each of these issues in turn.

Visual Display and Graphical User Interface Design

The consequences of this research for the design of visual displays and graphical user interfaces can be distilled into three guidelines to be considered when selecting an appropriate representation for a specific communicative goal.

First, designers of graphical interfaces should consider not only alternative representational formats but also how different quantities are encoded within any chosen format. Previous studies of display design have compared representations with distinct formats (e.g., Casner, 1991; Kosslyn, 1989; Zhang, 1996, 1997) or have analyzed the impact of different graphical representations of quantities (e.g., distance, shade, shape) on graph reading (e.g., Cleveland & McGill, 1984). In contrast, the representations in our study not only were informationally equivalent but also were similar forms of graphical representation requiring many of the same basic procedures to read. The primary difference between function and parametric graphs is in the way the quantity and argument variables are encoded using the Cartesian coordinate system. Nevertheless, significant computational differences, as evidenced by the RT data, were found between the two informationally equivalent (Larkin & Simon, 1987) graphs. This emphasizes the fundamental role and impact that representations have on cognition, as marked differences were found in the effectiveness of graphs that apparently have only small differences in representational terms.

Second, when selecting a graphical display, it is not sufficient to consider only the form of the representation in relation to a typical task (e.g., the AV-QV task in our study). The designer should address the full range of alternative varieties of the task (e.g., the QV-QV, QV-AV tasks). In the experiment the different encodings of quantities in the two graph types did not produce a uniform effect across all three types of question, as shown by the significant interaction between graph type and question type. This provides a clear illustration that when using graphical displays for apparently homogeneous tasks, such as reading off values, the fine-grained details of the task can affect performance.

Third, when choosing a representation, designers should not always select the one most familiar to target users but should attempt to balance the cost of familiarization with the computational advantages of less familiar representations. It may be the case that once an unfamiliar format has been comprehended, the computational advantages may be significant and may outweigh the cost of familiarization, particularly if the tasks are to be performed frequently or rapidly. In our experiment the magnitude of the effect arising from the particular combination of graph type and question type was not only significant but substantial. On two of the three question types, participants performed better using the parametric graphs by nearly a second in a task lasting about 5 s. This is particularly noteworthy, given that the participants were more familiar with the function graph. Hence for the task of reading off quantities, it appears to be better to use the less familiar parametric graph than a function graph. This challenges the common practice of using time series function graphs to present process data (e.g., Tufte, 1983).

Eye Movement Data

In a highly visual domain, such as graph-based reasoning, eye movements are an important source of information regarding the acquisition and processing of visual information during problem solving and the strategies people adopt when interpreting and working with graphs. This has been demonstrated previously by Carpenter and Shah (1998) in their analysis of eye movements in graph comprehension tasks, which revealed the cyclic nature of the pattern recognition and cognitive processes involved in graph comprehension.

GBR's eye movement predictions consist of an optimal sequence of fixations to achieve the goal based on the task analysis. When GBR is used, it can account for variations in aggregate RT data between users of different graph types (Peebles et al., 1999). By revealing the primitive actions of users at a very fine (20 ms) grain size, the eye movement study reported here, however, showed GBR's assumption of an optimal scan path to be an approximation that glosses over important cognitive and strategic factors at an individual level. The eye movement data also provide two additional constraints on the structure of the models, first by revealing behavior that can be interpreted as arising from working memory limitations or strategic decisions, and second by generating scan paths with which the model scan paths can be compared.

Cognitive Modeling

This research also clearly demonstrates the value of cognitive modeling in the analysis of complex interactive behavior. The diagrammatic reasoning tasks we have studied involve a complex interaction among three elements: the perceptual and cognitive abilities of the reasoner, the visual properties of the graph, and the individual requirements of the task. The GBR model of graph-based reasoning (Peebles et al., 1999) explains performance differences primarily in terms of the interaction between the latter two elements of this triad. Implementing the task analysis in a cognitive architecture has for the first time allowed us to take all three elements into consideration. By constructing computational process models incorporating cognitive factors (domain knowledge, problem-solving strategies, and working memory capacity) and perceptual-motor capabilities, we have been able to conduct a stringent test of the sufficiency and efficacy of the model by simulating every aspect of participants' behavior when carrying out the entire experiment.

The response latencies produced by the models provide reasonable fits to the data, and the overall pattern of relative task completion times resembles that of the observed data. This represents a major step forward from the account provided by GBR, which is limited to statements that the mean time to complete a given task will be greater for one graph type than for another. Moreover, the models' eye movement protocols and RTs emerge as a consequence of the learning mechanisms embodied in ACT-R – in particular the activation-based retrieval of declarative memory elements. This mechanism allows the model to capture various effects of practice on performance, such as the learning of symbol-variable associations and the locations of graph elements, and the subsequent forgetting of this information during a trial. This represents a critical aspect of the models' ability to capture the eye movement and RT data, and it is perhaps worth noting that this would not be possible using one of the GOMS class of task analysis techniques, such as CPM-GOMS (CPM stands for both critical path method and cognitive-perceptual-motor; John, 1990; John & Kieras, 1994) or UCIE (Lohse, 1993).

The cognitive models make a novel and significant contribution to the literature, therefore, by providing a precise and plausible account of the detailed sequences of saccades made by individuals in terms of strategic choice and the decay of base-level activation of perceptual chunks during the time course of problem solving. This level of detail in modeling visual attention in interactive tasks is still relatively uncommon (although see Byrne, 2001, Ehret, 2000, and Salvucci, 2001a, for other examples using ACT-R/PM; and Doane and Sohn, 2000, using Kintsch's, 1988, 1998, construction-integration model).

This research provides a clear example of the theoretical benefits that can be obtained by combining computational modeling using a cognitive architecture with detailed eye movement analysis. Specifically, this approach can begin to address issues regarding the cognitive factors underlying reasoning with external representations and make testable predictions of eye movements and individual task completion times. The current models represent a significant step toward the goal of integrating the three elements of the cognition-artefact-task triad in understanding complex interactive behavior and provide a basis on which to build models of more complex reasoning tasks with a broader range of diagrams.

ACKNOWLEDGMENTS

This research was conducted in the School of Psychology at the University of Nottingham and funded by a grant from the UK Economic and Social Research Council through the Centre for Research in Development, Instruction and Training.

REFERENCES

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies, 55,* 41–84.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The atomic components of thought* (pp. 167–200). Mahwah, NJ: Erlbaum.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Carpenter, P. A., & Shah, P. (1998). A model of the perceptual and conceptual processes in graph comprehension. *Journal of Experimental Psychology: Applied, 4,* 75–100.

Casner, S. M. (1991). A task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics, 10,* 111–151.

Chong, R. S., & Laird, J. E. (1997). Identifying dual-task executive process knowledge using EPIC-Soar. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (pp. 107–112). Mahwah, NJ: Erlbaum.

Cleveland, W. S., & McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association, 79,* 531–554.

Doane, S. M., & Sohn, Y. W. (2000). ADAPT: A predictive cognitive model of user visual attention and action planning. *User Modeling and User-Adapted Interaction, 10,* 1–45.

Ehret, B. D. (2000). Learning where to look: The acquisition of location knowledge in display-based interaction (Doctoral dissertation, George Mason University, 1999). *Dissertation Abstracts International: Section B: Sciences and Engineering, 60,* 5239.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47,* 381–391.

Gattis, M., & Holyoak, K. (1996). Mapping conceptual to spatial relations in visual reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22,* 231–239.

Gray, W. D. (2000). The nature and processing of errors in interactive behavior. *Cognitive Science, 11,* 205–248.

Gray, W. D., & Altmann, E. M. (2001). Cognitive modeling and human-computer interaction. In W. Karwowski (Ed.), *International encyclopedia of ergonomics and human factors* (pp. 387–391). New York: Taylor & Francis.

Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6,* 322–335.

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems* (pp. 107–115). New York: Association for Computing Machinery.

John, B. E., & Kieras, D. E. (1994). The GOMS family of analysis techniques: Tools for design and evaluation (Tech. Report CMU-HCII-94-106). Pittsburgh: Carnegie Mellon University, Human-Computer Interaction Institute.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12,* 391–438.

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

Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* Cambridge, England: Cambridge University Press.

Kosslyn, S. M. (1989). Understanding charts and graphs. *Applied Cognitive Psychology, 3,* 185–226.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11,* 65–100.

Lohse, G. L. (1993). A cognitive model for understanding graphical perception. *Human-Computer Interaction, 8,* 353–388.

Olson, J. R., & Olson, G. M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction, 5,* 221–265.

Peebles, D., & Cheng, P. C.-H. (2001). Graph-based reasoning: From task analysis to cognitive explanation. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 762–767). Mahwah, NJ: Erlbaum.

Peebles, D., & Cheng, P. C.-H. (2002). Extending task analytic models of graph-based reasoning: A cognitive model of problem solving with Cartesian graphs in ACT-R/PM. *Cognitive Systems Research, 3,* 77–86.

Peebles, D., Cheng, P. C.-H., & Shadbolt, N. (1999). Multiple processes in graph-based reasoning. In *Proceedings of the 21st Annual Conference of the Cognitive Science Society* (pp. 531–536). Mahwah, NJ: Erlbaum.

Pinker, S. (1990). A theory of graph comprehension. In R. Freedle (Ed.), *Artificial intelligence and the future of testing* (pp. 73–126). Mahwah, NJ: Erlbaum.

Savucci, D. D. (2001a). An integrated model of eye movements and visual encoding. *Cognitive Systems Research, 1,* 201–220.

Savucci, D. D. (2001b). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies, 55,* 85–107.

Schoelles, M. J., & Gray, W. D. (2000). Argus Prime: Modelling emergent microstrategies in a complex simulated task environment. In *Proceedings of the 3rd International Conference on Cognitive Modelling* (pp. 260–270). Veenendal, Netherlands: Universal.

Shah, P., & Carpenter, P. A. (1995). Conceptual limitations in comprehending line graphs. *Journal of Experimental Psychology: General, 124,* 43–62.

Tufte, E. R. (1983). *The visual display of quantitative information.* Cheshire, CT: Graphics.

Zhang, J. (1996). A representational analysis of relational information displays. *International Journal of Human-Computer Studies, 45,* 59–74.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21,* 179–217.

David Peebles is a senior lecturer in psychology in the Department of Behavioural Sciences, University of Huddersfield. He received his Ph.D. in cognitive science from the University of Birmingham in 1998.

Peter C.-H. Cheng is professor of cognitive science in the School of Cognitive and Computing Science, University of Sussex. He received his Ph.D. in artificial intelligence from The Open University, Milton Keynes, in 1990.

Date received: October 4, 2001
Date accepted: October 27, 2002