In Proceedings of ICCM - 2007- Eighth International Conference on Cognitive Modeling. 205 - 210. Oxford, UK: Taylor & Francis/Psychology Press.

A Qualitative GOMS Approach to Evaluating Diagrammatic Interfaces

B. Chandrasekaran and Tsviatko Yovtchev
Department of Computer Science and Engineering
The Ohio State University, Columbus, OH 43210 USA

Abstract

We describe an approach to evaluating diagrammatic schemes intended to support problem solving and decision-making. The methodology is in the GOMS framework in HCI, and is based on recognizing that the use of diagrams is part of a process that can be decomposed into a sequence of steps, each of which may be a Perception on the diagram, Inference, Transformation of the diagram and Visual Search. How well a diagrammatic scheme helps in a task depends on how well the human cognitive architecture can perform the actions in the various steps, and how the steps collectively contribute to performance measures such as time, error rates, and memory stress. We illustrate the approach by using it to analyze the use of some common data presentation displays in the task of discovering interesting relations between variables in a domain. Because of the current lack of quantitative empirical data about the execution of the basic operations by human architecture, the analysis is qualitative, which is nevertheless useful in providing useful insights. It also sets an agenda for empirical research to obtain the quantitative data needed, since the availability of such data would help significantly in evaluating and improving diagrammatic interfaces for decision support.

Introduction

The distinction between informational and computational equivalence (Larkin and Simon, 1987) of representations is relevant to explain the effectiveness not only of diagrams over text under appropriate conditions, but also of one kind of diagram over another for a specific task. Some diagrammatic schemes are lauded over others as especially well suited for specific tasks. This paper discusses an approach to evaluate interactive diagrammatic interfaces to support problem solving.

A diagrammatic scheme for a problem-solving task consists of specifications for representing information in a diagrammatic form, for meaningful perceptions on the diagram, and for modifying diagrams. When such a scheme is used to assist in problem solving, the problem solver engages in a *series of steps*, each of which may be an act of *perception* on the diagram, *inference* of new information by combining background knowledge and current information, including information given by an earlier act of perception, and *transformation*, i.e., modifying the diagram in some way to facilitate further problem solving. Information obtained by perception and inference will be added to the problem solver's short-term memory (STM), or added to the diagram in some form in a transformation step for pickup in a future perception step. The sequence of steps ends when the problem solver has acquired the information corresponding to the solution of the problem of interest, or gives up for whatever reason. Characterizing the problem solving activity in terms of these acts -- perception, inference, storage in memory, and transformation – arises naturally from a high-level view of the human cognitive architecture as comprised of central cognition, various perception modules and motor components.

Recognizing that diagrams are part of such a problem solving activity – instead of viewing them as stand-alone interfaces with one-way information flow from the interface to the user – is useful when we wish to compare alternative diagrammatic schemes for solving problems of a specified type. In particular, it makes possible the use of GOMS, the well-known analytical and comparative framework (Card, et al, 1983; John and Kieras, 1996). Our goal in this paper is to develop a GOMS-inspired analysis technique that is specialized for interactive problem solving with diagrammatic representations, and to illustrate it by applying into a set of alternative diagrammatic interfaces for a decision support problem in visual data mining. Another point of comparison is the work of Peebles & Cheng (2003). The measure of complexity is the number of perceptual attention shifts during a problem solving activity in comparing two graph representations for an information extraction task. As we'll see, we keep track of a more complex set of measures.

One evaluation measure of a diagrammatic interface might be the length – in total number of steps or time – it takes to solve an average problem or the hardest problem in a class. A variation of this measure might keep a count of each *type of step*, i.e., the numbers of perceptions, inferences and transformations, if the different steps correspond to different kinds of costs. Further distinctions may be made within each of these categories, e.g., some transformations and perceptions may have different costs than others. Another dimension of evaluation is propensity for error. A poorly conceived diagram might result in errors in one or more of the perception or transformation steps, or might overload STM and cause inference errors. Alternative diagrammatic schemes for solving a task may then be multi-criterially compared in terms of such measures.

Decision support systems (DSS) are an important class of applications of diagrammatic representations. There is usually a much greater role for transformations in DSS's than is normally the case in the use of simple diagrams. DSS's may be used in widely disparate circumstances: In one situation the cost of physical interaction may be high, with a corresponding preference for interfaces that do not require many transformations. In another situation, say where a user is multi-tasking, a DSS that entails many interactions would be preferable to one that places high stress on STM. Because of such wide variations in the conditions of use, a simple uni-dimensional measure of their performance, such as total time, is usually not adequate.

We introduce PTIS, a version of GOMS tailored to our needs. We then consider a set of alternate diagrammatic interfaces for a simple problem in data understanding: deciding if there are any interesting correlations between variables in a given domain of interest. The field of data mining focuses on problems of this type, and the interfaces we consider are often proposed as good visual presentations of data. However, the interfaces and the task are used as examples of the methodology, rather than the main subjects of the paper.

The PTIS Framework

When applying GOMS, operators are chosen that are generic enough to be applicable to the analysis of a variety of task/interface combinations of certain types. Two properties of an operator are especially important and are typically empirically obtained from studies on trained humans: the time it takes to apply the operator, and any error rates associated with application of the operator.

The operators we develop in our GOMS analysis of diagrammatic interfaces are at much higher grain sizes and complexity than the ones that GOMS research usually deals with (such as clicking buttons). For example, a basic operation in our domain is the perception of the best-fit straight line that approximates a cluster of data points. This is a common skill needed in experimental research. A person might need some training in this task, but once trained, he can visualize such a line. A corresponding motor operator is to *draw* such a line on a screen or on paper displaying such a cluster of points. Empirical data for time and error rates for operators at the grain sizes of interest to us are not yet available, so our current analysis is qualitative. However, the qualitative results are still useful in many situations, as we will demonstrate. When empirical data become available, it would be easy to convert the results to quantitative ones.

What is common in the use of all diagrammatic displays for decision support is that the user's actions belong to one of the following four types: Perception, Transformation, Inference, and (Visual) Search. The information obtained by Perception[1] or Inference is automatically placed in STM, i.e., it is not usually treated an operation. Since STM is capacity-limited, analysis should track STM load.

The operations in the Perception and Transformation categories respectively are chosen to represent basic units of the actions of the agent required for the task, but generic enough to be used as operations in a variety of tasks using diagrammatic interfaces. The best way to think of a Perception in this analysis is not as a gestalt perception act whose details the user does not access, but as a step in which the user is acquiring information from the display, and that the step has generality and reusability. Examples in data analysis are: visualizing a straight line that best summarizes a set of data; and visualizing the midpoint of a set of points. Because these Perceptions are general enough to be useful across a variety of data analysis tasks, investing in determining the timing and error rates associated with trained human perception would be worthwhile. (How these are learned would require a separate study.)

The relationship between Search and Perception needs clarification, since some of the Perceptions may also involve search. What we mean to capture in the Search category is the visual action needed to locate the objects that are the arguments for the specific Perception (and also for a specific Transformation). For example, on a display consisting of 50 labeled vertical bars whose lengths are proportional to the populations of 50 states, the Perception, `?Longer(bar x, bar y)`, would require the user to locate the two bars for the two states, and then apply the `Longer` Perception. Since comparing lengths of bars is an operation that would be useful for many tasks, and for which we can determine empirically the parameters for the human architecture, it is a good idea to separate this basic perception from searching for the items. The parameters for the perception operation would apply both to cluttered and uncluttered displays. Another reason to separate Search is that a visual search operation may also be applicable to Transformations, where a user may need to search and locate a button for a specific Transformation. In brief, we mean to include in Search visual search needed to identify the objects involved in given Perceptions and Transformations.

Inference is the name we give to the cognitive activity that processes the information in STM, by rule-based reasoning or mental imagery-like operations to obtain additional information. This process may involve additional elements brought from LTM to STM. Deciding on the next steps as well as solving the problem would typically require Inference steps.

In complexity the decision support tasks we consider occupy a place midway between using a display once to get some needed information, and open-ended interaction during which the steps are not pre-determined but require the user to engage in problem solving, e.g., to decide what Transformation should be applied next. We will assume in our framework that the user knows how to efficiently and effectively use the display for his purposes. This means that the tasks and the user's expertise are such that the next action to take is clear to the user, and that he has all the background knowledge needed for making the needed inferences. This is a fair assumption since displays for a task need to be compared based on intended optimal use of the display. (How hard it is to learn the best method for a display for a task is a separate issue, not dealt with here.)

Precision and Accuracy of the Various Operations. All the motor and perception related operations – Perception, Transformation and Search – are assumed to be potentially error-prone. For example, a user might make an error when required to choose the longer of two lines of almost equal length, or to distinguish between objects with very similar colors. It might also be hard to select a region with the mouse exactly within some planned coordinates, and finally, while searching among numerous items, one

[1] In the following analysis, we capitalize Perception, Transformation, Inference, etc when we intend to refer to operations that are to be taken as formally in the various sets of operations. We use lower case when we intend to refer to the general actions meant by the terms.

might miss the item that is sought or choose the wrong item. A heavy load on STM might result in loss of data, thus making the inference also unreliable. The PTIS technique allows for error rates, determined from empirical work, to be associated with the affected operation types.

Illustrative Task

We illustrate the approach by systematically applying it to a task that is common in data mining. The domain D of interest is characterized by a set of n numeric-valued variables $\{x_1, x_2, ..x_n\}$, and we have a set S of data about m entities in D. Then $S = \{d_1, d_2, ..d_m\}$ where $d_i = (x_{i1}, x_{i2}\ldots x_{in})$. We assume that the data are fully specified. Developing an understanding of the structure of D given S is a problem of great interest in data mining. A common form of such understanding is developing an account of any correlations that may exist between pairs of variables in $\{x_1, x_2, ..x_n\}$, and the ranges in which such correlations exist. The fact that correlations might exist over parts rather than the whole of the range makes visual means of hypothesizing such correlations especially useful, since standard correlation-detecting statistical algorithms might miss such correlations.

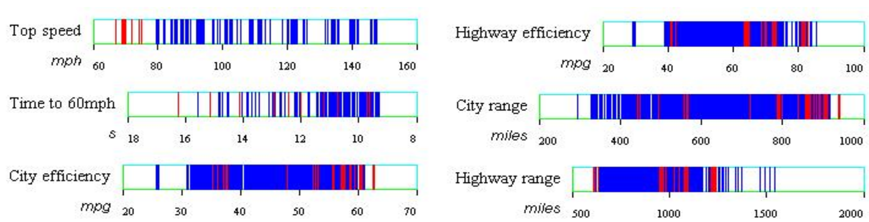

Fig. 1. The Spectrum display. (The figure has to be viewed in color.)

For example, in a case where there is a positive correlation over half the range and a negative correlation in the remainder, such algorithms would report no correlation at

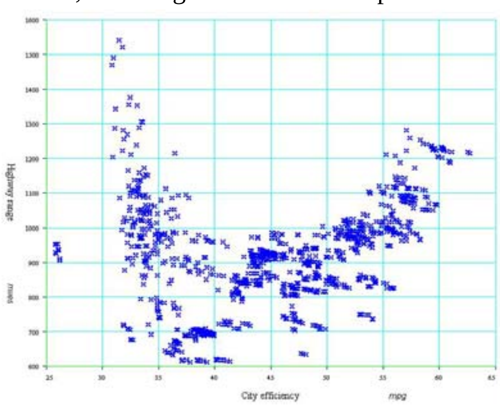

Fig. 2. Scatter diagram for two variables.

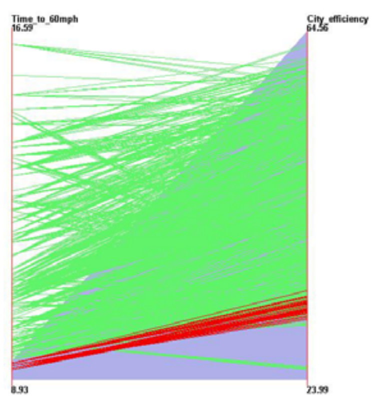

Fig. 3. Parallel Coordinates

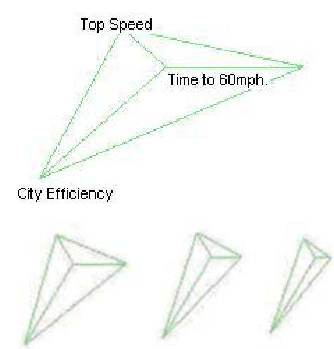

Fig. 4. Star Glyph; bottom, glyphs ordered by values of one of the variables

all. In contrast, a well-designed display (as we shall soon see) can help the user hypothesize such correlations easily.

A technical caveat is in order: such visual displays can only *suggest* correlations. The significance level of the correlation and the actual correlation coefficient can only be properly computed by statistical algorithms. We assume that once the user hypothesizes such correlations and ranges using visual displays, the data, the pair of variables *and the ranges* are input to an appropriate statistical algorithm. With the added information about the ranges, the algorithm can calculate the correlation parameters accurately. In the rest of the paper, we will focus on just the hypothesizing part of correlation discovery.

Though the more complex versions of the task raise additional interesting issues (see Yovtchev, 2005, for evaluation of the displays on the various versions of the task), in the available space, we will restrict ourselves to the simple version, below:

- *Task.* Given the set of data about some domain in the form of the values that m entities from that domain take on two variables x_1 and x_2, hypothesize all the subranges of the variables x_1 and x_2 where the correlation coefficient differs from 0.

Diagrammatic Displays Considered

A number of diagrammatic forms have been proposed to represent data of the type we described[2]. In this paper, we use a subset of these displays – sufficient to introduce the approach and make the main points.

Spectra. In this display (Fig. 1), each variable is represented by a horizontal strip – the strips are typically normalized so that their ranges take up approximately the same length – and each of the entities in the data set is represented by a vertical stripe (the height of the stripe has no significance) in the default color, say blue, at the location corresponding to the value of the entity on that variable. More than one entity may have the same value for a variable, so the entities might be *stacked* at that location, and when entities are *dense*, i.e, many are close together, they may not be visibly distinct.

[2] For Spectra and Scatter plots, we used the Viewer (Josephson, *et al*, 1998), available from Aetion Technologies LLC, www.aetion.com. For Parallel Coordinates and Star Glyphs, we used XMDV tool (Ward, 1994).

Transformations. The user can select a window of variable size on any of the spectra (e.g., the window [18-16] in the "Time to 60" Spectrum in Fig. 1). This changes the color of the entities in the window (in Fig. 1 they appear in red), not only in the Spectrum where the window was selected, but on all the other Spectra as well. (As a new window is selected, the old window is automatically cleared.)

Scatter Diagrams. This display (example in Fig. 2) is a 2-axis Cartesian graph, with one variable on each axis. For n variables, a maximum of $n*(n-1)/2$ scatter plots are possible. The entities are represented as points at locations corresponding to their values on the variables. Remarks we made on stacking and density of entities in the Spectrum case apply here as well.

Transformations. The user can select a rectangular window in any of the scatter plots, and the entities in the window will change color, not only in that scatter plot, but in all other scatter plots that are constructed.

Parallel Coordinates. This display (Fig. 3) has n parallel axes - one per variable displayed. The m alternatives are displayed as m paths of $n-1$ straight-line segments crossing the axes at positions corresponding to the entity values in the respective variables (Fig. 3 shows just two variables). Remarks on stacking and density that we made earlier also apply for this display.

Transformations. The user can select a range in any of the variables, and the entities in the selection window, and the lines connecting the values on other axes of each of these entities will change color. Fig. 3 shows a selection.

Star Glyphs. Each entity is represented as a glyph, which consists of n rays (for n variables) going out of its center whose endpoints are connected to form a polygon, as in Fig. 4 which shows an example for 3 variables. The length of a ray is proportional to the value of the entity on that variable. Making a glyph requires a minimum of three variables. The bottom of Fig. 4 shows a Star Glyph display of 3 objects, each represented on 3 variables, and ordered by the values on the variable on the ray at 0^o. There are no Transformations available.

Performance Analyses

Sizes of Selection Windows. Many of the methods call for making selections using a window, whose size the user needs to set. First, the window size needs to be large enough to capture enough samples so that the hypotheses are statistically meaningful. The size also determines the smallest range over which meaningful correlations may be hypothesized. If the window size is say 10% of the range of the variable, then any changes in correlations in ranges of the same order cannot be detected. There are precise statistical formulas available (Yovtchev, 2005) to make these estimates.

Spectrum Display

Method:

1. Transform display by making a selection window over the range of x_1 of size at most half as large as the smallest subrange over which any existing correlation is to be detected, where the window starts at the beginning of the range of x_1.
2. Perceive and store in memory the midpoint (mean) of the resulting selection in x_2.
3. Transform display by defining another selection window of the same size as the first one, but beginning where the first one ended. (We assume in analysis that the current window is automatically cleared.)
4. Compare the midpoint (mean) of the resulting selection in x_2 with previous k memorized ones. Infer and remember the trend that resulted from the comparison, and the beginning of the range where the trend emerged, or Infer the end of a trend that has been present so far. (Here by "trend" means whether the midpoint moves to the right or left systematically as the window moves to the right, or whether the midpoint movement has no systematic connection to the direction of the movement of window.)
5. Repeat the procedure until the end of the range of x_1 is reached.

Perception: *Perceiving midpoint of a set of points*. In this Perception the user mentally estimates the midpoint of a given set of points on a line, e.g., the midpoint of the set of red points in the "Highway Range" Spectrum in Fig. 1. This activity has an associated error measure. It may involve sequential mental computations, but it is useful to treat it for our purposes as a reusable unit of mental activity.

Transformation: The only Transformation operation is window selection. In some display versions, a window may need to be explicitly deleted; in that case, the number of Transformations will double.

Inference: *Determining the trend in mid-point position*. This action can be modeled in finer detail as keeping the previous k midpoint locations in STM, and comparing their values, to determine if a trend, positive or negative exists, and if there was a trend, whether it continues or has stopped. The higher the value of k, the more reliable the estimate, but higher also the load on STM.

Analysis: The sequence of operations is as follows: Select window (Transformation) in x_1, Perceive midpoint in x_2, Clear-and-Select next window (Transformation), Perceive midpoint, ..Infer trend in direction of midpoint, Transform, Perceive, Infer..

Without making finer distinctions, for a first approximation, the process takes r/s Perceptions, Transformations, and Inferences, where r is the range of x_1 and s is the size of the selection window measured in the same units as the range.

The maximum load on STM would be ($k + 3$), the sum of the number of previous perceptions over which the trend is inferred, and a pair of locations and one sign (positive or negative) for each of the correlation ranges discovered.

Errors. In addition to the intrinsic error rate in the Perception of the midpoint, the display adds another potential for error: since more than one entity can occupy the same point, the user has no immediate access to the density information, and the midpoint estimate might be skewed. Because of missing density information, and due to the inherent human error in the basic perception involved, correlations may be missed, and even when detected, the starting and ending points could be off by some amount.

The load on STM, which can be quite high, can also lead to errors due to data loss. The error in tracking the direction of the movement of the midpoint due to STM load can be decreased by the user revisiting earlier window locations and repeating the movements, but this is at the cost of an increase in the number of Transformations.

Ideas for Display Improvement. The big source of error, *viz.*, potential high stress on STM, can be minimized by changes in display design. If the user had access to a Transformation whereby beginning and end of each hypothesized correlation range can be marked on the screen along with its sign, STM load would be minimized. However, this would increase the total number of Transformations by 3 per correlation range. Further trade-offs between STM load and increase in number of Transformations are possible.

Scatter Diagrams

Method: In the case of 2 variables, there is only one scatter diagram, as in Fig. 2. The scatter diagram is the most direct way to perceive any correlations. It calls for Perceiving correlation regions directly, as one can see in Fig. 2, that there is a negative correlation from x_1 value of 30 to 45, and a positive correlation from 45 to 60.

Perception, Transformation and Inference: Perceiving plausible regions of correlation can be modeled as cluster detection, where the clusters are characterized by scatter around a straight line, perceiving the beginning and end points of the straight lines and the sign of their slopes. The task calls for distinguishing between clusters whose axis has a slope of 0 from those with a non-0 slope. Subject to confirmation from empirical data, it seems to us that correlation hypothesizing in this case is much less error-prone than in the Spectrum case – no stressing of STM; also faster, since it is direct and skips those inference steps that are needed for the Spectrum display. Nevertheless, in comparison with the optimal algorithms, there are bound to be some errors in the precise location of the end points, and also there are potentials to miss and mis-hypothesize correlations with a low correlation coefficient. Assuming that the hypotheses generated are to be fed to mathematical algorithm to generate quantitative information about the correlation, users might be trained to err in the direction of hypothesizing correlations when they are doubtful, with the idea that the numerical procedures might be able to reject dubious hypotheses.

There are no Transformations or Inferences needed for the 2-variable case, and thus there is little load on STM.

Analysis: With the proviso that the act of Perception described above is complex, involving a sequence of mental operations, the task simply calls for one act of Perception. The temporal complexity of this Perception is approximately linear in the number of correlation regions, with a minimal part that would exist even when there are no correlations.

Errors. As mentioned, there are inherent errors in human perception of correlation, the error increasing as the correlation coefficient decreases. There are also errors in the locations of the end points. We hypothesize that both these errors are inversely proportional to the number of entities, i.e., human performance would have less error as the number of entities increases.

Display Improvement. An additional Transformation, Zoom, might help if applied for repeated Perception operations to locate the end points. Of course, this change to the display design will increase the number of Perceptions and Transformations for completing the task.

Because of its simplicity, low error rates and low stress on STM in comparison to the alternatives, the Scatter Diagram can be taken as the gold standard display for the task under consideration.

Parallel Coordinates

Method: Depending on the density of the data, different methods seem to be appropriate.

Relatively Sparse Data. When the density of the data is low enough that the values that an entity takes on different axes can be distinguished, the correlation regions, if any, and the directions of the correlation are available for Perception.

Relatively Dense Data. In this case, since lines connecting the values of the individual entities cannot be distinguished (Fig. 3), the method is similar to that for the Spectrum display. A variation on this method is track the average slope of the lines created by the window (the average slope of the red lines in Fig. 3). If the slope starts and stays positive (negative) within a region, positive (negative) correlation may be hypothesized.

Perception, Transformation and Inference: In the case of Sparse data, the basic Perception is not gestalt as in cluster recognition in the case of the Scatter Diagrams, but involves a sequence of comparisons. The user sweeps through a range of x_1 and visually follows the slopes of the lines connecting to x_2. Thus, it is likely to take longer time, and is possibly more error-prone. In the Dense case, once selection is made, the required Perception is similar to that in the Sparse case, and the same remarks apply. For Sparse Data, as in the case for Scatter Diagrams, there is no need for Transformation and Inference operations; for Dense Data, remarks made in the case of Spectrum display apply.

Analysis: For Sparse, except for likely higher error rates in Perception, the same analysis as for Scatter Diagrams applies. For Dense, remarks similar to that in the Spectrum case apply, with possibly different values for error rates for Perception.

Design Ideas*:* The error analysis of the cases coincides respectively with the Viewer's Spectrum and Scatter Diagram error analysis. Hence, it leads to the same design ideas -- zoom functionality, and markup operations to mark starts and ends of hypothesized correlations. The latter trades off STM overload for an increase in the number of Transformations.

Star Glyphs

Since the glyphs have to have a minimum of 3 variables, we add a pseudo variable whose values are the same for all the entities (or it is a real variable in the domain whose values we ignore). Let us also assume that the glyphs are ordered on their values on x_1 as in the bottom part of Fig. 4.

Method: Scanning the glyphs in order of the value of x_1, for glyph i, compare its x_2 value with k previous values to infer whether a trend of increase or decrease has begun, and if already begun, maintained. If the trend just began or ended, save the value of i to STM. Continue until all the glyphs are scanned. The value of k is set based on considerations described when we discussed inference in the use of the Spectrum display, i.e., to smooth out random local variations. As before, higher k reduces statistical error, but errors due to resulting overload of STM might reduce or eliminate the advantage.

Perception, Transformation and Inference: The basic Perception is one of local comparison of x_2 values to decide if an increase or decrease is observed. There are no Transformation operations. The issues regarding Inference are similar to our corresponding discussion for the Spectrum case. That is, the x_2 values of k glyphs are kept in STM, and their values are compared to determine beginning, maintenance or end of positive or negative covariation trends. As before, the higher the value of k, the more reliable the estimate, but higher also the load on STM.

Analysis: The number of basic Perception steps is (m-1), the number of glyphs. The number of Inference steps is (m – k). Maximum load on STM is (k + 3* number of correlations ranges hypothesized), since each correlation region requires remembering 2 end points and its sign. Because no windows are used to average out behavior, the numbers of Perception and Inference steps are quite large.

Error rates. The basic Perception is quite reliable, except when the increase or decrease is very small, in which case the error does not likely matter much. The Inference step is error-prone because of the complexity of calculation, and the requirement on STM to keep k items. Starting and end point assessments are especially likely to be error-prone because of natural statistical variations on x_2 values, which need to be smoothed out during the Inference step.

Design Ideas. As before, the load on STM may be reduced by providing Transformation operations to mark the beginnings, ends and the signs of the correlations.

Comparing Displays

Even this level of qualitative analysis is useful in making comparisons. The Scatter Diagrams are the most direct – no Transformation operations, no Inference, and little stress on STM. The Glyphs are especially laborious to use, and the Perception and Inference steps seem prone to high error rates for both the Glyph and Spectrum displays. Whatever the general attractiveness of the Glyph displays, they are not well suited for the specific task we considered.

Concluding Remarks

The paper outlines an approach in the GOMS framework to systematize investigating how good specific diagrammatic schemes are for specific families of tasks. Unlike earlier applications of the GOMS framework, which involved elementary operations at a relatively low level of granularity, diagrammatic interfaces used in decision support systems involve relatively complex perceptions and physical interactions. We illustrated the approach by a comparative performance analysis of several candidate diagrammatic interfaces for the task of discovering relations between variables in some domain of interest. The analysis results in estimates of the numbers of various basic operations, such as Perception, Transformation, Search and Inference, and of stress on short-term memory. For many DSS applications qualitative results as we obtained are sufficient. The precise timings about how long the entire process would take may be less important than whether the display calls for significantly more interaction compared to another display, whether perceptions are more likely to be error-prone in one than another, etc. However, empirical data about the timing and error rates of the human cognitive architecture on the basic operations can be used for more precise predictive evaluations. We expect to launch such an initiative soon.

The approach can help to identify aspects of the display that need improvement. Adding Transformations to mark partial results on the display may be considered if the analysis indicates potential for STM stress. If analysis indicates that the contribution of errors in specific Perception is significant, alternatives might be considered.

Goodness of an interface given the best method is not the same as how good it is in helping someone *learn* the best method. Our methodology can be applied to the latter task as well, and it is an important future direction of research.

Acknowledgments

This paper was prepared through participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009.

References

Card, Stuart, Moran, Thomas P., and Newell, Allen (1983). *The Psychology of Human-Computer Interaction,* Lawrence Erlbaum Associates, Hillsdale, NJ.

John, Bonnie E. and Kieras, David (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4):320-351, December 1996.

Josephson, John R., Chandrasekaran, B., Carroll, Mark, Iyer, Naresh, Wasacz, Bryon, Rizzoni, Giorgio, Li, Qingyuan, and Erb, David A. (1998). An architecture for exploring large design spaces. In *Proc. of the National Conf on AI* (AAAI-98), pages 143-150. AAAI Press/The MIT Press.

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

Peebles, D., & Cheng, P. C.-H. (2003). Modeling the effect of task and graphical representation on response latency in a graph reading task . *Human Factors*, 45, 28-46.

Ward, Matthew O. (1994). XMDV tool: Integrating multiple methods for visualizing multivariate data. In *IEEE Visualization*, 326-333, 1994.

Yovtchev, Tsviatko (2005). PTIS - A High Level Framework for Comparative Evaluation of Decision Support Interfaces, MS Thesis, Computer Science and Engineering, The Ohio State University.