Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1987 Proceedings

International Conference on Information Systems (ICIS)

1987

AN EMPIRICAL STUDY OF STRATEGIES FOR UNDERSTANDING QUANTITATIVE DECISION MODELS

Charles Wiecha
Carnegie Mellon University

Max Henrion
Carnegie Mellon University

AN EMPIRICAL STUDY OF STRATEGIES FOR UNDERSTANDING QUANTITATIVE DECISION MODELS

Charles Wiecha
Max Henrion
Department of Engineering and Public Policy
Carnegie Mellon University

ABSTRACT*

Demos is a modeling environment designed to help a co-operating team design, analyze, critique and refine quantitative models for policy research. Earlier research found that readers of Demos models tended to become disoriented while exploring models online. In response we have designed and implemented a graphical interface to Demos named Demaps. Demaps displays diagrams of the model structure, both dependence networks and abstraction hierarchies, to provide graphic context and direct manipulation style of interaction. We describe a study of the use of Demaps to understand and compare multiple versions of models. The study employs verbal protocol analysis to evaluate the design of Demaps and to discover expert strategies for model understanding and criticism. Subjects were able to learn to use Demaps effectively in about an hour to review and compare policy models and perform sensitivity analyses. The study describes two strategies used in reading models and suggests the desirability of additional facilities for recording model critiques and accessing detailed background information on models.

*We gratefully acknowledge the contributions of many people, including Jill Larkin, Jim Morris, Granger Morgan, Andrew Appel, and our subjects. This work was supported by the Information Technology Center and the National Science Foundation under grant IST-8316890.

INTRODUCTION

Increasingly, quantitative models are being used to try to illuminate complex questions of technology, risks and public policy. One example, which we shall use as an illustration, is policy towards installation and use of seat-belts, airbags, and other automobile safety systems (Graham and Henrion 1984). A few other examples are health and air pollution, the disposal of radioactive waste, "acid rain," and nuclear weapons. Such models typically require an integration of scientific and technical information, forecasts and value judgments. To evaluate seat-belts and airbags, one needs technical data on their effectiveness in reducing fatalities and injuries, projections of costs and usage rates, and judgment of the amount of investment appropriate to save a life or injury. Information and judgments may be obtained from a variety of experts, including biomedical engineers, highway engineers, economists, social scientists, and policy makers. The resulting models may be reviewed by various interested parties, such as automobile manufacturers, insurance companies, and drivers' organizations. There are typically considerable uncertainties about technical data and forecasts as well as value judgments. In such cases, the purpose of quantitative modeling should not be to obtain definitive numerical results, which are anyway unattainable, but rather to integrate information, serve as a focus for debate, and provide insights about what issues and uncertainties are (or are not) critical.

Computer implementations of policy models have frequently served to impede rather than support this co-operative process of review and debate. Such models have conventionally been written in FORTRAN, or other procedural languages, with documentation of model assumptions and data in separate hardcopy reports. Inadequate documentation tends to prevent extensive validation, review and refinement by others than the model author. "Black box" models, in which model assumptions and structure are hidden, do not encourage confidence in results and inhibit the participation of external reviewers. The resulting models are liable to have little impact in policy formation unless perhaps as a spurious justification commissioned to support preconceived positions.

Demos (Decision Modeling System) was conceived as a tool to alleviate some of these problems (Henrion and Morgan 1985). It provides a flexible, non-procedural modeling language, which defines the mathematical relationships between variables, leaving it to the system to worry about flow of control and sequence of execution. Non-procedural models are much more compact and more easily understood by non-programmers than conventional procedural languages (Henrion et al. 1986). Demos supports interactive creation, examination, analysis and modification of models. It allows representation of uncertainty by ranges of alternative values and probability distributions, with general facilities for parametric sensitivity analysis, uncertainty analysis and automatic graphing of results. It employs Monte Carlo and related techniques for propagating probabilistic values. Documentation and text explaining what variables and their relationships represent are included as attributes of variables as an integral part of the model representation. A large model may be organized as a hierarchy of submodels, with controlled scope and access between submodels. Demos also allows the creation, management and comparison of multiple versions of a model. For more details see Henrion and Nair (1982) and Henrion and Morgan (1985).

Over several years, Demos has been applied to several dozen policy problems of a variety of types and complexity. Many of these projects have been designed and refined by multidisciplinary teams. For example, an integrated assessment of the acid rain problem involved about ten people, including specialists on pollution control technology, atmospheric chemistry, long range transport, aquatic impacts, materials effects, economics, and uncertainty analysis (Marnicio et al. 1985). Experimental studies and informal evaluations of the use of Demos have shown several advantages over more conventional approaches to policy modeling, particularly the ease of building non-procedural models and the ease of representing and analyzing uncertainties (Henrion et al. 1986). However, initial studies of the use of Demos for online reviewing and critiquing policy models were somewhat disappointing. Reviewers tended to get disoriented and lost when exploring even quite small models, taking only a few screens to display. They would often resort to printing out the whole model in an effort to get a global perspective of its structure. Similar problems of disorientation were reported in studies of ZOG, a menu-based network of text-frames (Mantei 1982).

Partly to deal with these problems, we designed a graphics-based interface, named Demaps, which provides graphical diagrams (maps) of the model structure both to provide context and to support interaction by direct manipulation for browsing and editing models. This paper focuses on one of a series of experimental studies intended to evaluate the use of Demaps and to examine how expert modelers comprehend and critique a model. An examination of what information users find helpful in understanding models can lead to better displays, documentation, and automatic model explanation systems. This study identifies the information required by users at each stage of model understanding. The stages include reading documentation and values, forming questions about model structure and behavior, constructing commands for sensitivity analysis, and drawing conclusions based on model results. By categorizing the problems associated with each stage, we suggest detailed ways in which the modeling language and interface can be improved.

Following a description of Demaps, we will outline the experimental procedure and analysis of results and describe some results in terms of a preliminary model of comprehension and critique. In conclusion, we will discuss some of the advantages of using such an interactive *softcopy* representation for model scrutiny instead of a traditional hardcopy text, and some of the problems identified in this study.

DEMAPS: A GRAPHICAL DECISION SUPPORT SYSTEM

The four mechanisms of abstraction, hierarchical decomposition, multiple views, and connectors between multiple views function together to reduce disorientation in large Demaps models. Figure 1 shows a view of a model on the costs and benefits of passive restraints in reducing automobile fatalities. Individual variables are shown as nodes shaped like ovals in the diagram. Each node has two connection points, one below the node for links from variables it depends on, and the other above the node for links to variables it in turn influences. Thus data flows from the bottom to the top of the diagram in Figure 1. The diagram is an abstraction of the concrete model in that only the dependencies between variables are shown graphically. Many different functional forms would be represented by the same set of nodes and links. The detailed way in which each variable depends on others is specified in its algebraic definition. The definition is not visible directly in the diagram, but in the scrolling text and pop-up displays shown in Figure 1.

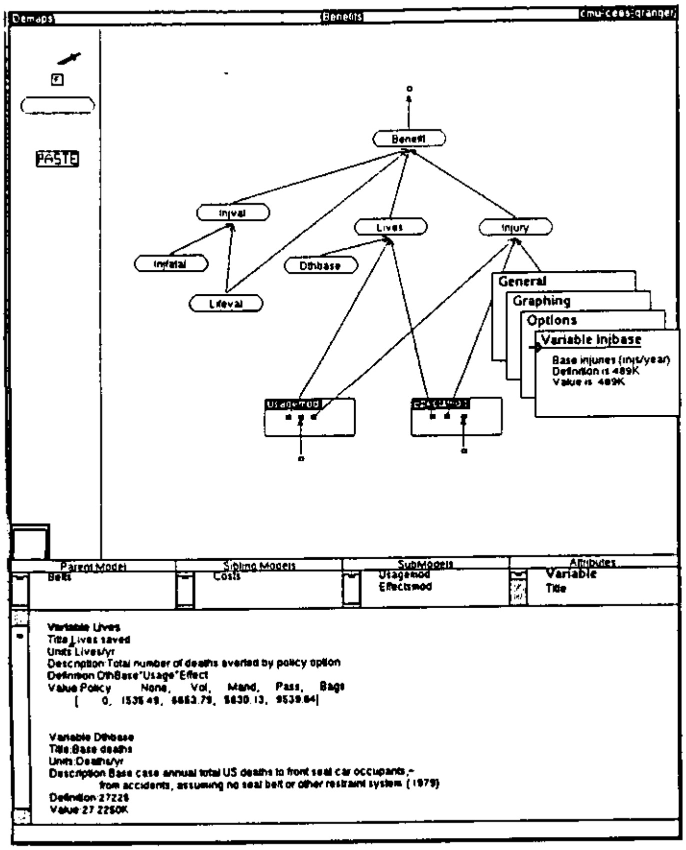

Figure 1. Detailed View of Benefits Portion of Automobile Restraint Model in DEMAPS. Variables are shown as ovals and links between them indicate the flow of data. Variables external to a submodel are shown by small square connector nodes. Menus attached to node give abbreviated textual descriptions of each variable.

The abstraction of Demaps diagrams is important in understanding and designing models. The lack of an explicit representation for algebraic operators in the diagrams allows models to be read and designed in stages which consider each of the following concerns independently:

- o What are the significant variables that should be, or are, included in a model, and which lie outside of its scope?
- o What are the qualitative dependencies among the variables included in the model?
- o What are the quantitative algebraic definitions which implement the dependencies?

Thus the first two stages of the design of quantitative decision models involve *qualitative* choices about significant variables and their interrelationships. These choices can be made in practice as follows. As each variable is added to the diagram by dragging an oval icon from the column at the left of each diagram, a blank template of attributes is created in the text display. As links are drawn graphically to other variables, its textual definition is automatically modified to list those variables on which each variable depends. Since the graphical links are an abstraction of the definition, it cannot automatically specify the actual functional form of the resulting dependencies. Demaps creates a **FunctionOf** relation to represent the abstract dependencies in text form. For example, if links were drawn from a variable X to variables A and B, the definition of X in the text display would be set automatically to: FunctionOf(A,B). Links may be removed between variables graphically by using the dagger icon from the left column. As each link is cut, the FunctionOf definitions are altered to reflect the new set of dependencies.

Demaps does not require understanding and design to proceed linearly from stage to stage. Rather, the abstraction of model structure facilitates each type of consideration without overly constraining the later stages. By focusing attention on different considerations at each stage, the diagrams can be an important aid in structuring debate about alternative model designs. Like the idea graphs of Cognoter (Foster and Stefik 1986), influence diagrams in Demaps help make model structures transparent and invite others to comment on and revise them.

Models may be decomposed into a hierarchy of *submodels* which are shown as boxes in Figure 2. Submodels have both *external* and *internal* views. An external view, for example of the benefits submodel in Figure 2, displays the interface between the submodel and its external context. The external view consists of a box representing the submodel along with connections to variables which are inputs to, or depend on outputs from, the submodel. The internal view of the benefits submodel shown in Figure 1 displays the details of the submodel implementation by showing the variables and connections relating the submodel's inputs to its outputs. Submodels are a second form of abstraction in Demaps diagrams which allow models to be built and viewed by hiding information about components which are not relevant in a given context.

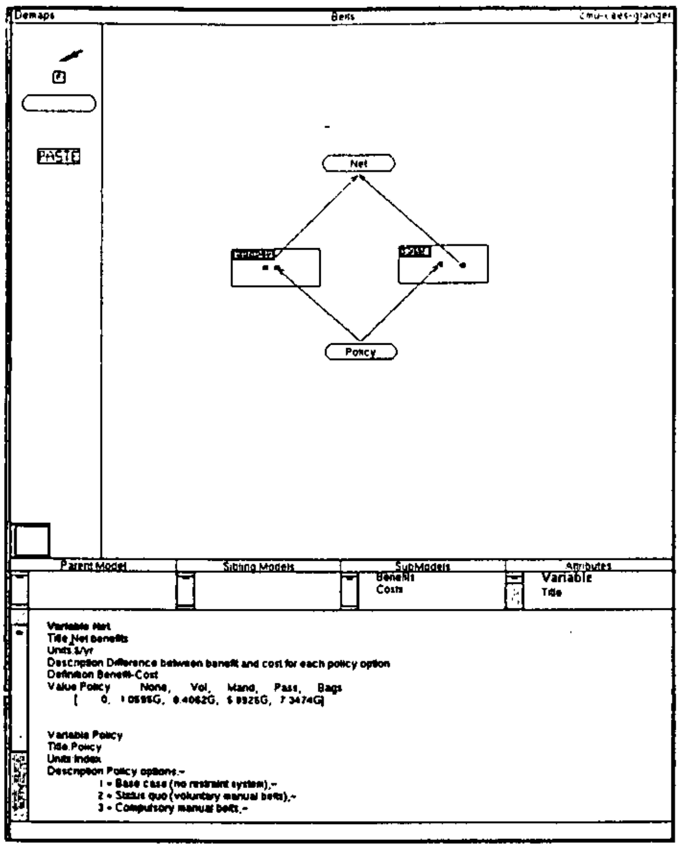

Figure 2. The External View of the Model's Benefits and Costs Computations

The hierarchical set of submodels are displayed using the control panels at the center of Figure 2. Each panel contains the names of the displayed model's parent, sibling, and child submodels. The current model view can be shifted to any of these other submodels by clicking on one of their names, or multiple displays created to view several model diagrams simultaneously.

Linkages between variables in different submodels are represented by offpage connectors, shown as small squares in Figures 1 and 2. Offpage connectors may link a variable to variables in other models as in Figure 1, or they may link models to each other when the outputs of one model are used directly as inputs to another as in Figure 2. In both cases, menus can be displayed when the mouse is positioned over a connector node which lists the remote variables represented by that connector. The structure of the diagram is thus a form of fish-eye view (Furnas 1986) in that details of a limited area of the system are selectively augmented with those objects at a greater "distance" from the focus of attention which are significant given the current view.

METHODOLOGY

Black, Galambos, and Reiser (1983) distinguish between verification and discovery research. Verification research typically involves controlled statistical tests intended to confirm or reject hypotheses. Discovery research, such as described in this paper, is used in a new area of research to reveal broad patterns of novel behavior and to suggest hypotheses for future testing. Controlled experiments typically cannot record the wealth of information necessary to suggest how an interface functions to support model understanding. For these reasons, we and other researchers are beginning to apply the paradigms of information processing psychology to the study of programmers (Pennington 1982) and programming environments (Soloway 1984; Anderson et al. 1984). We employ "thinking aloud" verbal protocols (Ericsson and Simon 1984) to supplement observations of user interactions with the system and to build models of behavior in this task.

TASK

We asked subjects to compare two versions of a cost-benefit analysis model of various policies towards seatbelts and to select one policy. The

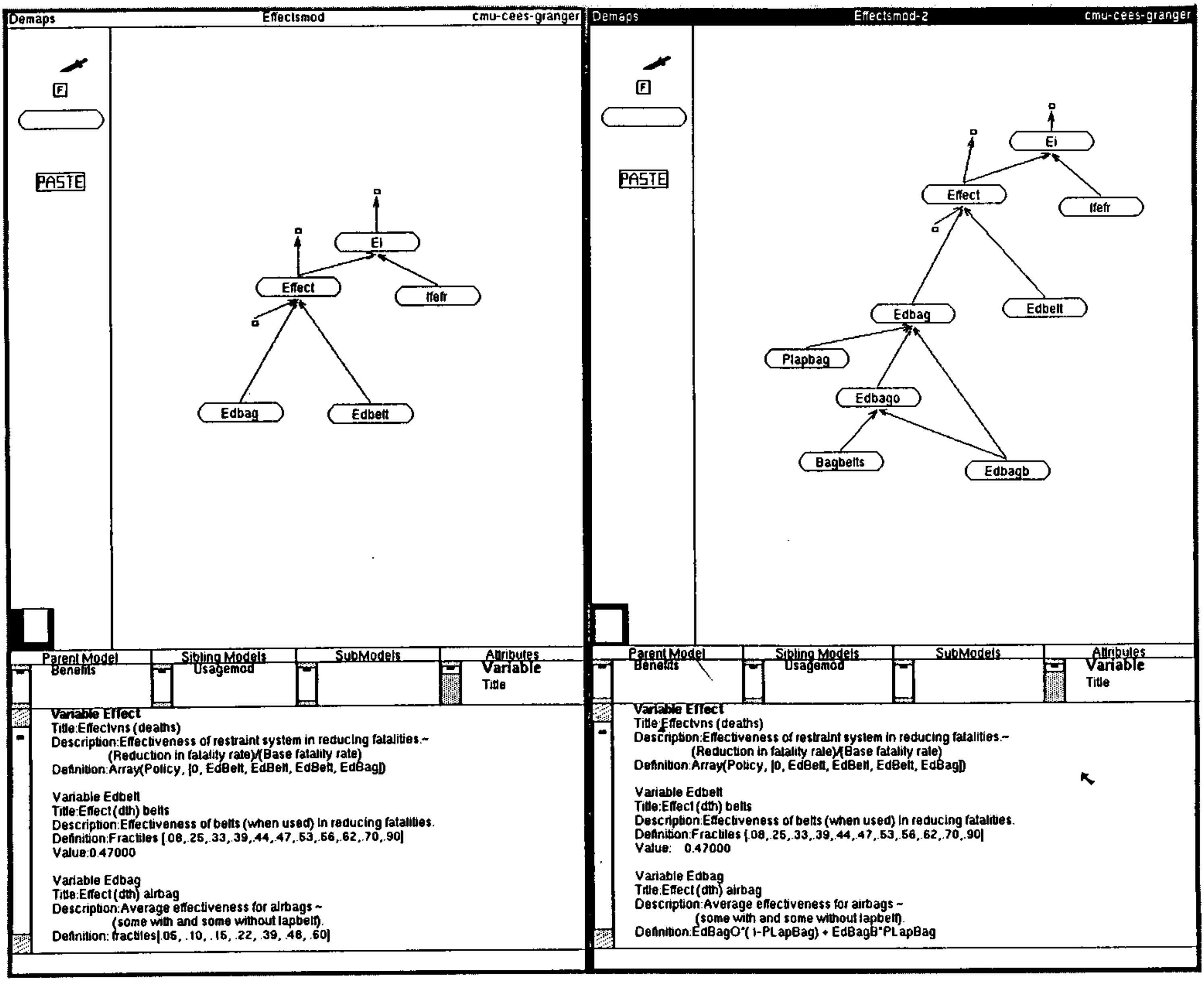

Figure 3. Side by Side Display of Alternative Model Versions. Plots of variable values appear on demand and are located below the corresponding model.

task of comparing and critiquing alternate versions of a model is important in any co-operating modeling project with multiple participants. The refinement of models in a cooperative setting often is driven by reviews of previous work by other group members. In this study, the same base model version is compared with a different alternative version in two sessions for each subject. Each session lasted less than an hour. The three task instructions are listed below. The policies considered by each model included no seatbelts, a law mandating their use, and combined belts and airbags.

- o You are a policy analyst at the US Department of Transportation (DOT) charged with deciding whether to require passive automobile restraints on new automobiles manufactured for sale in the United States. To help you make this decision, you have subcontracted a review of the costs and benefits of three policy options to two separate research firms.

- o You have before you the results of their modeling efforts. Please examine the two models, decide which policy each of them recommends, and select one option for implementation based both on the evidence in the models and on your general knowledge policy analysis.

- o While performing this task, please think aloud. The experimenter is not permitted to answer any questions about the models. Thank you.

In the study, Demaps displayed both model versions side by side on the screen, each with a diagram and text window as shown in Figure 3. Since the two versions had different implications, subjects were required to explore and understand them in depth in order to evaluate which, if either, was most appropriate. They were able to conduct "what-if" sensitivity analyses to examine the importance of any assumptions they found suspect. To reduce the time spent identifying model differences (which the system should do automatically in any event) subjects were given a list of variables which differed between the two versions.

Four faculty and postdoctoral students at Carnegie-Mellon University were selected as subjects. While an objective measure of expertise is not available in decision modeling, subjects were chosen based on their experience in developing and evaluating large scale quantitative policy models. We studied expert, rather than novice, modelers so that our results might be used in developing guidelines for novices on effective modeling techniques. Each subject was given a separate training session prior to the study to learn the use of Demaps, with an informal test before each study session to make sure they were familiar with all commands. The number of subjects was limited by the time required for transcribing and analyzing each protocol.

PROCEDURE

Each session was recorded on video tape using three cameras, one each on the subject's face, the screen, and a note pad. Using synchronized images from the screen display and subject's face, it was possible to determine the model window and panel (diagram or text) the subject was looking at for each phrase in the protocols. The analysis of these data are presented in a separate paper (Wiecha and Henrion 1987). Subjects were asked to think aloud as they worked and could make notes on the pad of paper provided next to the workstation. The only role of the experimenter was occasionally to prompt the subject to keep talking.

DATA ANALYSIS

The audio track of each videotape was transcribed and separated into phrases. The video track of the tape was transcribed to associate a command or area of the display with each phrase. Each phrase was classified as an instance of one or more of a set of *elementary* cognitive processes (Wiecha 1986). The elementary processes typically involve reading short passages of model documentation from the screen or drawing immediate inferences from that documentation. Short sequences, or episodes, of the elementary processes were grouped into one of the eight *comprehension* processes listed in Table 1. A sample passage from one protocol is given in Figure 4.

To judge the consistency of this coding process one of the eight transcripts (that judged as most difficult to encode originally) was completely recoded after a delay of three weeks. The recoded transcript was compared with the original coding and any occurrence of extra, missing, or differing processes counted as an error. The recoded and original transcripts agreed in 91% of 480 elementary processes.

The elementary processes are used to study how attention shifts between variables as the subject glances from graphical to textual model representations (Wiecha and Henrion 1987). The comprehension processes, described in this paper, show the progression of model understanding from syntactic details (identifying version differences), to model structure and behavior, finally to policy recommendations based on extensive sensitivity analysis. The results are used to suggest ways in which Demaps and other decision support systems can better support the understanding and communication of decision models.

MODEL COMPREHENSION PROCESSES

This study is concerned only with the relationships among the aggregate comprehension processes listed in Table 1 and not with the details of their internal composition. Figure 5 shows how attention shifts among the processes. Data in Figure 5 are aver-

```
257 yea we get worse
      [ComprehendBehavior
               [ReadValue(Netbenefits) [ReadTextRight(Netbenefits)]]
               [InferStatic(Netbenefits)]
258 so in general the tax does not seem to improve enforcement
               [InferTrend(Enforcement)]
259 sufficiently to increase overall netbenefits
               [PropagateTrend(Netbenefits)]
260 ah enough to justify having any tax at all
               [PropagateTrend(Tax)]
      ]
261 now I'm going to go back to enforcement here
      [ComprehendStructure
               [SelectVar(Enforcement) [PopUpSelectMethodRight(Enforcement)]]
262 which is defined as
               [WindowControl [ScrollUpRight]]
               [WindowControl [ScrollUpRight]]
               [WindowControl [ScrollDownRight]]
               [WindowControl [ScrollDownRight]]
               [WindowControl [ScrollUpRight]]
263 scrollbar is a pain
               [Meta]
264 ratio of enforcement tax to total cost
               [ReadDescription [ReadTextRight(Enforcement)]]
      ]
:
:
```

Figure 4. Fragment of Coded Protocol. Elementary processes such as ReadValue and InterStatic are nested within comprehension processes such as ComprehendBehavior and ComprehendStructure. Protocols are automatically analyzed to generate transition probabilities.

aged across all subjects. Individual differences in the data are discussed in detail below.

Each node in Figure 5 represents one of the processes listed in Table 1. Each successive pair of *elementary* processes is counted as a transition. A pair of elementary processes may both lie in a given comprehension process or originate and terminate in different processes. Transition probabilities are the ratio of the number of transitions along each link to the total number of transitions originating at each node.

Across all subjects, there were a total of 3,207 transitions. Where the computed transition probabilities fall below 0.01, the corresponding links have been suppressed in the diagram. Thus while there are no links drawn into, or out of, INSTRUCTIONS and OBJECTIVE, these processes are nonetheless observed in some subjects.

The total number of elementary processes in each comprehension process has been given both before and after the first What-If analysis in each session. This division is used to reveal how transition patterns which occur while subjects are mainly focused on understanding the two models differ from those which occur while subjects are testing their understanding. The separate totals reveal that comprehension processes are clustered into two groups as described in the next section. Taking each elementary process as an equal unit of effort, the totals also give an estimate of the effort expended in each comprehension process.

Two Strategies for Reading Models

The average transitions in Figure 5 describe a *two stage* model, in which all of the version differences are identified by DIFFERENCES and understood by STRUCTURE and BEHAVIOR beforany sensitivity

Table 1. Comprehension Processes

- **Read Task Instructions** [INSTRUCTIONS]
  - INSTRUCTIONS is indicated by phrases in which the subject reads the task instructions aloud or silently.
- **Identify Model Differences** [DIFFERENCES]
  - DIFFERENCES analyzes the two model versions to identify those variables which differ between them. DIFFERENCES is concerned only with identifying the differences, not with understanding how or why they are different.
- **Understand the Current Structure of a Model** [STRUCTURE]
  - STRUCTURE is concerned with understanding the given structure of a model or to understand how the functional form of parts of two versions are related.
- **Understand the Behavior of a Model** [BEHAVIOR]
  - BEHAVIOR is concerned with understanding the behavior of a model individually, with understanding the differences in behavior between two model versions as currently given, and with understanding the results of sensitivity analysis.
- **Generate Candidates for Sensitivity Analysis** [GENERATE]
  - GENERATE is indicated by phrases which choose variables for examination by the other sensitivity analysis processes.
- **Formulate Sensitivity Test** [COMPOSE]
  - COMPOSE structures a Demos expression to be used as part of a sensitivity analysis.
- **Consider an Alternative Objective Function** [OBJECTIVE]
  - OBJECTIVE is indicated by phrases which reconsider the model's objective function, and possibly suggest another one for analysis.
- **Make Policy Recommendation** [RECOMMEND]
  - RECOMMEND is indicated by phrases summarizing information gathered so far about the model in support of one or more policy decisions.

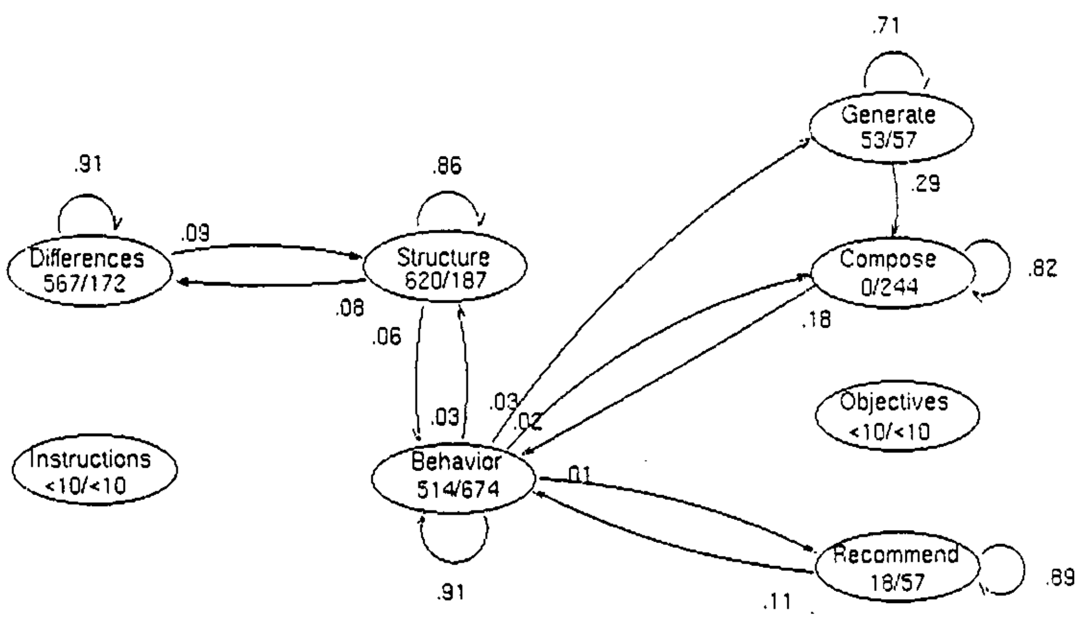

Figure 5. Average Transactions Among Comprehension Processes. Transitions represent fraction of successive elementary processes which occur within and between each comprehension process. Figures before (after) "/" are number of elementary processes before (after) the first What-if test.

analysis is done. This pattern is indicated in Figure 5 by a cycle between DIFFERENCES and STRUCTURE before sensitivity analysis, which shifts to one involving GENERATE, COMPOSE, and BEHAVIOR during sensitivity analysis. Subjects with this shift are said to use a READ-BEFORE-TEST strategy in that they understand each model well before experimenting with it.

In three of the eight sessions, however, there are many DIFFERENCES and STRUCTURE processes *after* the first What-If test. These subjects are using an incremental strategy which defers examining model documentation until required later on. We call this strategy TEST-BEFORE-READ.

Three of four subjects (B, C, and D) each use the TEST-BEFORE-READ strategy in one of their two study sessions. In many cases, subjects using the TEST-BEFORE-READ strategy re-examine a model's structure in response to unexpected results from a What-If test. In such cases, the subject has formulated an expectation of the outcome of the test but finds conflicting results from the model. Several successive incorrect hypotheses related to a single variable may be tested before an effort is made to comprehend the model more completely. These results suggest that the TEST-BEFORE-READ strategy may be less effective than reading the model more completely from the start.

On the other hand, the READ-BEFORE-TEST strategy may not always be feasible. In large models, particularly, there is often too much material to be completely read. The important research question is how can we make the TEST-BEFORE-READ strategy more effective? In small models, the entire structure of the model can be internalized. Meaningful questions can then be asked since information about the interactions between each variable and the rest of the model is accessible directly in the reader's memory. One way large models can be made more understandable is by attaching *constraints* to the value of each variable. Warnings are generated automatically whenever a variable violates its constraint. The constraints themselves are a type of documentation which can be used to develop an appropriate understanding of the purpose of each variable. Another approach is to allow users to annotate models with comments about each variable, as described below.

Role of Critiques in Understanding Large Models

In this study, subjects seemed to have no preconceived notion of which variables should be questioned in the models. This is perhaps surprising, since a number of variables in the study models are typically controversial, such as the "value of life." Rather, subjects select each What-If test based on the results of prior tests and on knowledge gained during earlier explorations of the model. This knowledge is summarized by a number of explicit model critiques which identify variables and clusters of variables which have been found to be problematic or are of interest for further exploration.

The importance of model critiques as a source of ideas for sensitivity analysis can be seen in Table 2. In three of four subjects, between 80% and 90% of all variables selected for sensitivity analysis (by the GENERATE process) had been the focus of a prior critique. Most critiques occur when questioning the model behavior (in BEHAVIOR processes) but a number also occur when generating ideas for sensitivity analysis (in GENERATE).

Table 2. Percent of Varaibles in What-If Tests that have been Mentioned in Critiques

SUBJECT	N	PRIOR CRITIQUE
S2	9	.89
S4	4	.50
S5	5	.80
S6	11	.91
Average	7	.83

Critiques thus emerge as a major organizing factor in model comprehension strategies. They summarize knowledge gained from exploration of the model and suggest ideas for further exploration. Critiques may thus provide a means for improving the effectiveness of the TEST-BEFORE-READ strategy which stresses incremental learning resulting from successive sensitivity analyses.

The central problem with TEST-BEFORE-READ was formulating sensible What-If tests without completely understanding a model. One way to gain a better understanding of small parts of large models may be to annotate them with critiques attached to individual variables. Other reviewers could browse through a set of critiques previously attached to a variable while formulating their own ideas for sensitivity analysis. Through exposure to previous comments, reviewers would understand each variable well enough to question it intelligently without having explored the rest of the model.

The process of reading and writing critiques constitutes an ongoing dialog among multiple model reviewers. Such a dialog emphasizes the view of models as *flexible* structures whose purpose is as much to encourage debate as to provide numerical outputs for direct use in decision making. In teaching, we often see that beginners view modeling as a rather linear process (Henrion et al. 1986). Models are written, implemented in a system such as Demos, then run perhaps only once to produce results. Most experts see modeling, however, as an iterative process involving considerable experimentation and sensitivity analysis. Indeed, all of our expert subjects spent over half of their time in sensitivity analysis during our study sessions. By engaging students in an ongoing computer-based discussion of models through stored critiques, we are optimistic that they will more quickly adopt the iterative strategies typical of experts.

Need for Iin-depth Model Documentation

One very interesting and frequent critique in our data is that Demaps lacks sufficient information to understand models even for subjects using the READ-BEFORE-TEST strategy. Subject A, for example, was bothered by a lack of information on how estimates of the usage and effectiveness of seatbelts had been elicited from experts. Subject D was unable to completely explain how a negative net benefit is related to the assumptions behind a particular function in the model. Subject E abandoned the task altogether, stating that he was unable to proceed without significantly more information about the model's assumptions than was given. His session was not transcribed and coded and does not appear in any of the other results, but was significant for its emphasis on the need for easy access to in-depth model descriptions. Other subjects, such as B, had other types of critiques which could be satisfied by additional information about the design assumptions and data collection procedures implicit in the models. Clearly, for four out of five subjects, more details than are currently provided are required in order to fully evaluate the behavior of models.

The key question here is whether it is appropriate to attach all such information as attributes of variables and submodels, or whether the somewhat different kinds of organization possible in linear text reports have intrinsic advantages. The model-based organization may work better for organizing explanations and criticisms when there are many modelers and reviewers. Certainly organization around the mathematical model structure makes it clear just what information and explanation is available or lacking for each component, which may not be so clear in a linear text.

In related work on electronic books ("hyper-text") (Brown University 1986) there is a notion of a *web* of links among chunks of text. Multiple versions of the webs may be created to allow a given database to be browsed according to linear or network organizations. Further experimentation on applying these ideas to quantitative decision models seems called for here. However, it is relevant to note Borenstein's finding, in his study of Help systems, that the quality of the text was much more important than the access mechanism (Borenstein 1985).

CONCLUSIONS

We have used Demos and the Demaps graphics interface to develop several hypotheses about the advantages of a computing environment to support interactive design, analysis, and critique of decision models. The advantages of an interactive "softcopy" computing environment over the traditional hardcopy report as a medium for reviewing and critiquing a decision model include the following:

- o Understanding of the model may be obtained by *active* exploration of model structure and behavior rather than *passive* reading of text in a fixed sequence.
- o When finding suspect assumptions, reviewers may immediately perform sensitivity analyses to see whether or not they are likely to matter.
- o Automatic facilities for sensitivity and uncertainty analysis can support reviewers in identifying and prioritizing critical assumptions and uncertainties.

- Since documentation and explanatory text is integrated with the mathematical model, it is much easier to maintain consistency between text and model as the model is refined and multiple versions are developed.

Earlier studies of Demos revealed that many of these advantages were curtailed, since users tended to become disoriented and lost when exploring even relatively simple models. Demaps appears to successfully cure these problems by providing hierarchical influence diagrams to give context and allow easy browsing. Our subjects proved able to learn to use Demaps effectively in an hour or so.

The study reported here revealed three areas where further experimentation with Demaps should be conducted. First, we observed two strategies for reading models. In one, subjects seek to understand a model completely before experimenting with it. The other is an incremental strategy in which subjects let the results of each experiment drive the acquisition of further information from the model. We believe that the latter strategy is likely to be used frequently when exploring large models. Further research is needed on mechanisms for indexing documentation to make the strategy more effective.

Second, improved facilities are required for recording model criticisms and other comments as attributes of variables and submodels, annotated according to the authors and model version. This is particularly important where a team of several model builders, reviewers and decision makers are jointly engaged in a co-operative process, and they need to record and communicate their opinions to one another.

Third, most subjects felt that the information supplied in the model was incomplete and inadequate. Indeed the information contained in the experimental model on policies towards seatbelts was quite limited, much less than appeared, for example, in its published descriptions (Graham and Henrion 1984). Of course, any representation of a decision model is liable to such criticism, be it hardcopy or softcopy, given the inevitable variability among model builders and reviewers about how much and what type of background information and explanation is necessary. The research question is how such information can be effectively integrated into a modeling environment. Ideally, a single documentation database could be viewed both as a linear document (suitable for hardcopy reading) and as a network of descriptions attached to specific parts of a model (suitable for interactive browsing).

The subjects used in this study were experts in quantitative policy modeling and were skilled in reviewing and identifying possible weaknesses in such models. But such skills are relatively rare, and all too often important models are not exposed to thorough scrutiny. A model development process involving thorough review, debate and iterative refinement by a team with varying perspectives is much more likely to lead to a model that can engender confidence and play a significant role in policy formation. To the extent that advanced modeling environments can facilitate such a process, they may contribute to more effective use of such models. However, to be able to involve a wider range of participants with less developed skills, such environments should offer more active support for reviewing models, explaining in verbal form non-intuitive model behavior, and identifying potential deficiencies. Currently we have only a rough idea of what the skills of expert modelers are; but, by means of such studies as this, we hope to understand them better, with the eventual goal of incorporating some of this expertise into the modeling environment to make it more accessible to non-experts.

REFERENCES

Anderson, J. R.; Boyle, C. F.; Farrell, R.; and Reiser, B. "Cognitive Principles in the Design of Computer Tutors." *Proceedings of the Cognitive Science Program*, 1984, pp. 2-11.

Black, J. B.; Galambos, J. A.; and Reiser, B. J. *Coordinating Discovery and Verification Research.* Erlbaum, 1983.

Borenstein, N. *The Design and Evaluation of On-line Help Systems.* Unpublished Ph.D. Dissertation, Carnegie-Mellon University, 1985.

Brown University. "Intermedia." *Proceedings of the Conference on Computer Supported Cooperative Work*, Austin, TX, December 1986.

Ericsson, K. A., and Simon, H. A. *Protocol Analysis: Verbal Reports as Data.* MIT Press, Cambridge, MA, 1984.

Foster, G., and Stefik, M. "Cognoter, Theory and Practice of a Collaborative Tool." *Conference on Computer-Supported Cooperative Work*, Austin, TX, December 1986, pp. 7-15.

Furnas, G. "Generalized Fisheye Views." *Human Factors in Computing Systems, CHI'86 Conference Proceedings*, April 1986, pp. 16-23.

Graham, J. G., and Henrion, M. "The Passive Restraint Question: A Probabilistic Analysis." *Risk Analysis*, Vol. 4, No. 2, 1984.

Henrion, M., and Morgan, M. G. "A Computer Aid for Risk and other Policy Analysis." *Risk Analysis*, Vol. 5, No. 3, 1985, pp. 195-208.

Henrion, M.; Morgan, M. G.; Nair, I.; and Wiecha, C. "Evaluating an Information System for Policy Modeling and Uncertainty Analysis." *American Journal of Information Science*, Vol. 37, No. 5, 1986, pp. 319-330.

Henrion, M., and Nair, I. *DEMOS User's Manual*. Department of Engineering and Public Policy, Carnegie-Mellon University, 1982.

Mantei, M. *A Study of Disorientation Behavior in ZOG*. Unpublished Ph.D. Dissertation, University of Southern California, 1982.

Marnicio, R.; Rubin, E.; Henrion, M.; Small, M.; McRae, G.; and Lave, L. "A Comprehensive Modeling Framework for Integrated Assessments of Acid Deposition." *Proceedings of the Air Pollution Control Association Annual Meeting*, June 1985.

Pennington, N. *Cognitive Components of Expertise in Computer Programming: A Review of the Literature*. Technical report, Graduate School of Business, University of Chicago, July 1982.

Soloway, E. "A Cognitively-Based Methodology for Designing Languages/Environments/Methodologies." *Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, ACM Special Interest Groups on Software and Programming Languages, April 1984, pp. 193-196.

Wiecha, C. *An Empirical Study of How Visual Programming Aids in Comprehending Quantitative Policy Models*. Unpublished Ph.D. Dissertation, Carnegie-Mellon University, 1986.

Wiecha, C., and Henrion, M. "Linking Multiple Program Views Using a Visual Cache." *Interact-87 Proceedings*, 1987.