Intelligent Support of Secondary Data Analysis

Russell G. Almond
Educational Testing Service

1.0 Software Support for Data Analysis

There is little doubt that computers have had a large impact on the way data are analyzed. Not only has the computer opened the door for new analytical techniques (such as interactive graphics and automatic model selection) but it has also made statistical analyses available to a much larger class of users. Statistics packages must increasingly meet the needs of analysts with limited formal training in statistics, and thus must be able to provide advise and guidance in addition to calculations.

Looking at how statisticians make decisions about statistical methods and techniques reveals that they consider both data and Meta-Data—information about the Data. (See e.g, Hand [1993,1994]). This presents a difficulty for designers of statistics packages: much of this meta-data must all be entered into the computer before the computer can give meaningful advise. In many cases, this meta-data is available only off-line (and some times it is not even recorded). Without one key part of the meta-data—how to extract the data from a database or collection of datafiles—it is impossible to analyze the data.

Secondary analysis of a large public data source requires the transmission of the key meta-data from the primary to the secondary analyst. In a large public survey, such as the National Assessment of Educational Progress (NAEP), the primary analysts who do the data collection and original analysis under contract from the government are not the ultimate users—the educational researchers and policy analysts who are trying to understand the factors that influence student achievement. The secondary analysts don't have ready access to the meta-data, rather they must extract it from the printed manuals and research reports. Storing this meta-data on-line, in a form which is accessible to analysis software would not only help secondary analysts find the relevant meta-data, but also enable the analysis software to offer advise and make intelligent default decisions.

This paper describes the NAEP-VUE system, a proposed system for analyzing data from the National Assessment of Educational Progress. This system is still in the early prototype phase, so this paper mostly lists features of the proposed system, especially ways in which it can incorporate meta-data. Feedback, particularly additions to this list of meta-data types, are most welcome.

2.0 The NAEP Survey

The National Assessment of Educational Progress (NAEP) is a comprehensive, ongoing study of multiple aspects of primary and secondary education. Over its 25 year history, it has been continually adapted to meet the diverse needs of both policy makers and educational researchers. To achieve high accuracy estimates of small sub-populations, NAEP employs a complex multi-staged sampling design using both stratification and clustering. To maximize the breadth of information while maintaining comparability across years and minimizing the burden on individual students, items are administered in balanced incomplete blocks which create complex patterns of missing data.

In order to analyze the NAEP data, researchers must understand (O'Reilly *et al.* [1996]):

1. The definition and meaning of the variables. In some cases this involves obtaining the original survey question or test item. In other cases it involves understanding the meaning of derived and scaled scores, item reliabilities and other properties of the variable.
2. How to extract the data relevant to their research hypotheses from the database. The program NAEPEX (Rogers [1995]) addresses some of these issues, but it is not well integrated with the data analysis tools.
3. How to account for the complex sample design in the analysis. For most analyses this is an issue of choosing between the supplied weights, but it also could involve deeper limitations on the variables.
4. The multiple imputation procedure and its limitations. The NAEP public use data supply multiple "plausible values" for items missing due to non-response, survey design as well as latent variables (proficiencies). The analysis software must be able to summarize over the plausible values (Almond

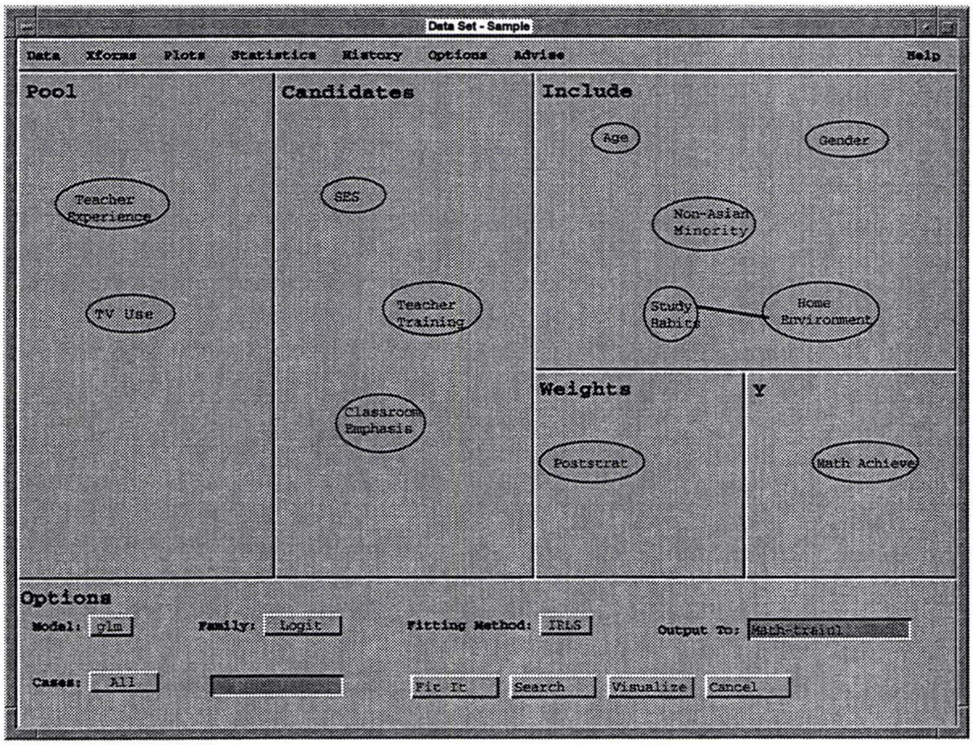
and Schimert [1995]). Furthermore, the analyst should avoid analyses using interactions which were not included in the imputation model.

5. The level of analysis appropriate for each variable. Some variables are collected at the student level, some at the school level and some at the state level. In many cases the appropriate analysis is a hierarchical linear model (Raudenbush [1988]).
6. Confidentiality issues. If the data are too finely divided, then it could be possible to identify single individuals or schools.

The researchers need access to this information as they are doing the analysis within an integrated system for data preparation, analysis and interpretation (Riddle, Fresnedo and Newman [1995]).

3.0 The NAEP-VUE Environment

Most statistical packages focus the user's attention on the procedures used to fit the model rather than what the data and the model are trying to tell the analyst about the underlying educational process. While this may be appropriate for a statistician, whose job is to develop new statistical methodology, it is inappropriate for a data analyst, whose job is to gain understanding of the process being studied. The alternative suggested by Anglin and Oldford [1994] is to focus on the model (and the data) as a vehicle for user interaction. In this spirit, the main focus of user activity in NAEP-VUE will be the model specification dialog (Figure 1). This dialog is motivated by the graphical model representation of statistical models (Thoma and Goodall [1991], Whittaker [1990]).



Model: glm
Family: Logit
Fitting Method: IRLS
Output To: Math-traidl
Cases: All

Figure 1. NAEP-VUE *Model Specification Interface*

This is a mock up of the model specification interface. The example shown in the figure is motivated by Jacobi and Stevens [1996]. By dragging the variable icons (nodes in the graph) the analyst can specify which factors are to be definitely or possibly included in the model. By connecting the variable icons the user can specify which interaction terms to include in the model. The options panel give access to other model fitting options and case selection. Visualizations, transformations and other data manipulations are available through the menus.

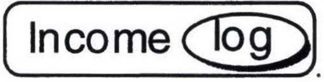
Through this dialog the secondary analyst should be able to specify either a single candidate model or a space of candidate models. In addition to the screen regions, the interface will use color and line and background shading patterns to indicate which interactions are definitely included, which are candidates, and which are excluded. NAEP-VUE will take the graphical description of the model, convert it to a model formula (in the Wilkinson and Rogers [1973] formula language), extract the data necessary to fit that data, export it to a data analysis package for fitting, and report the results back to the user. The results will be available as a "fitted model" object for use in later analyses and visualizations.

Once a model is (partially) specified, it becomes the focus for a number of other interactions, including visualizing raw-data, model fits and diagnostic statistics from the model. Because the computer knows the analysts intentions (through the model dialog) it can make meaningful recommendations in these areas.

3.1 Model Specification Interaction

Most model specification tasks can be accomplished by dragging variable objects to various areas of the screen. Those areas are:

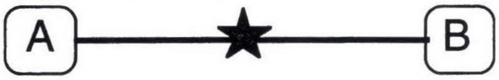
- Pool — Repository for defined variables. Dragging derived variables back into the pool saves their definition in the database for later use.
- Y — specifies response
- Weights — specifies weights
- Include — force in model (e.g., effect that are to be tested)
- Candidate — available for model selection (e.g., possible control variables).
- Selection (not shown) — Indicator variable describing cases to be included.
- Grouping (not shown) — Categorical variable specifying separate analysis for each level in group (or possibly overlapping shingles as in Cleveland [1993])

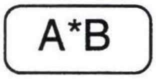
Transformation of the variables will be available from a variety of pull-down and pop-up menus. NAEP-VUE will allow the analyst to specify both transformations and candidate transformations to appear in model selection procedures. These will appear as decorations on the variable icon, 

.

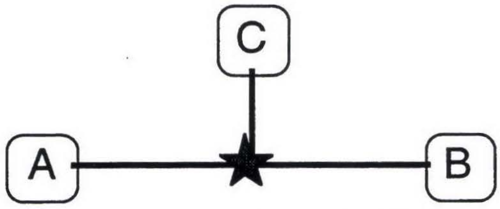
Interactions are indicated either by connecting the variables or by separate interaction nodes (this allows the user to specify definite inclusion for the variable but only candidate inclusion for the interaction terms. Nesting is show by nesting the variable icons as shown below.

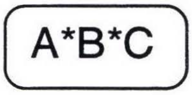
- Two way



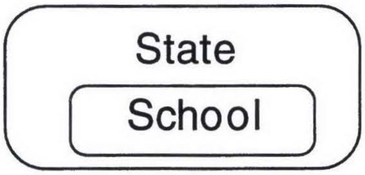


- Three way





- Nesting shown by nesting



Similarly, the analyst can create "one of these" sets by grouping the variables. This technique allows the user to specify that only one possible transformation of a variable, or only one of a set of collinear variables should be included. This supports the idea that variable transformations should be considered within the model selection framework (Faraway [1992]). Note that the analyst specifies a collection of candidate models, this is consistent with the philosophy of model uncertainty (*e.g.*, Madigan and Raftery [1994]) and could be used to promote those ideas.

Parts of the model which are not easily specified by dragging variables, are specified through the dialog below. This allows access to features such as model class and fitting algorithm.

4.0 Variable Type Hierarchies

The variable icons in the selection dialog represent more than columns of numbers, they point to rich objects containing both data and metadata. In NAEP-VUE, variables are represented as objects within an *ontology* (Gruber [1991]). Important information about a variable, such as the survey question, which level it was collected on, and the primary analysts notes about the variable are attached as properties of the variable. However, the variables will be part of an object hierarchy relationship, similar to the one prototyped in Almond[1995]. This allows, the primary analysts to attach "advisor" operators at high levels which will be appropriately inherited. For example, an advisor which suggest square root transformations could be attached to the variable class "count." Hand [1993] suggests other variable type based statistical advise and Roth *et al.* [1994] suggests selecting visualizations based on the variable types.

Variable type hierarchies aren't new, there are many programs which can now take advantage of the most basic of types hierarchies (Figure 2). For example, the New S program (Chambers and Hastie [1992]) distinguishes between continuous predictors and factors (which are automatically coded with dummy variables) when fitting a linear model or generalized linear model and chooses contrasts for dummy variables based on whether or not the factor is ordered. The program JMP (SAS Institute) chooses the model type based on both the type of the predictor variable and on the type of the responses. This simple use of functional polymorphism (dispatching the function on the types of the arguments) reduces the program specific knowledge needed by the use to operate the program. The user learns a single syntax for the single fit model command instead of separate commands for each model type (whether accessible via command line or menu, this is a large reduction in user memory requirements).

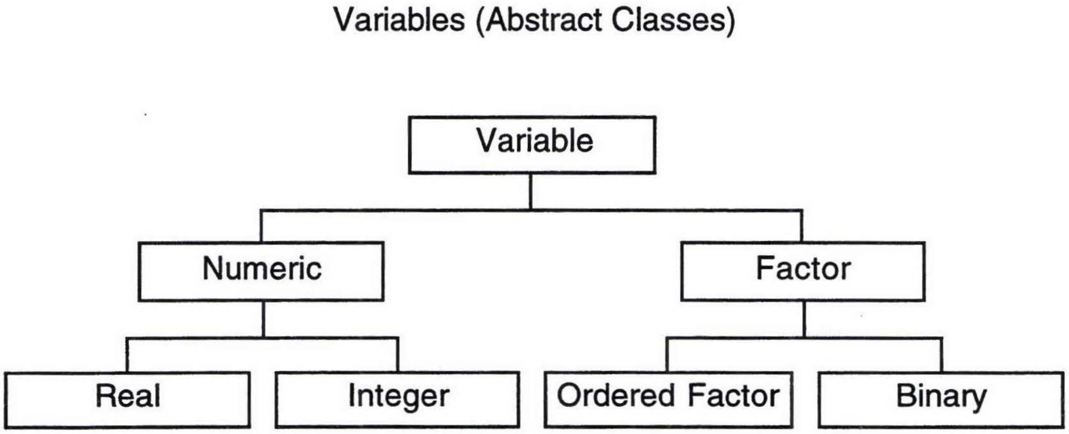


Figure 2. Top level of Variable Hierarchy

However, this simple dispatching only scratches the surface of what can be done with a full variable hierarchy (Figures 3 and 4). Mosteller and Tukey [1977] (Chap 5) give some general advise on transformations based on the type of the variable. For example, Mosteller and Tukey suggest that for a counted fraction, transformations which a symmetric around 50% (folded fractions) are often useful, where other types of counts and amounts are often better re-expressed using logs and square roots. Possible transformations should be suggestions, not automatically applied. Transformation advise can be expressed in several ways, including messages and the use of defaults and ordering in menus.

The variable type hierarchy helps organize meta-data about the variables. Meta-data attached in high places in the hierarchy and is inherited at lower levels unless it is specifically overridden. For example, the recommended transformation for an *amount* variable type might be logs (then square roots), but the *time*

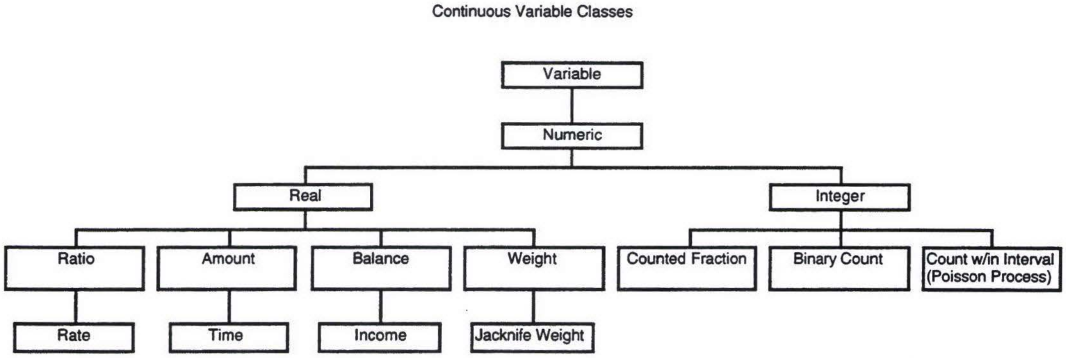


Figure 3. Variable Hierarchy for Numeric Types

subtype might override that with a recommendation to look at reciprocals first. In NAEP-VUE, the primary can add specific recommendations for specific variables (when appropriate and available).

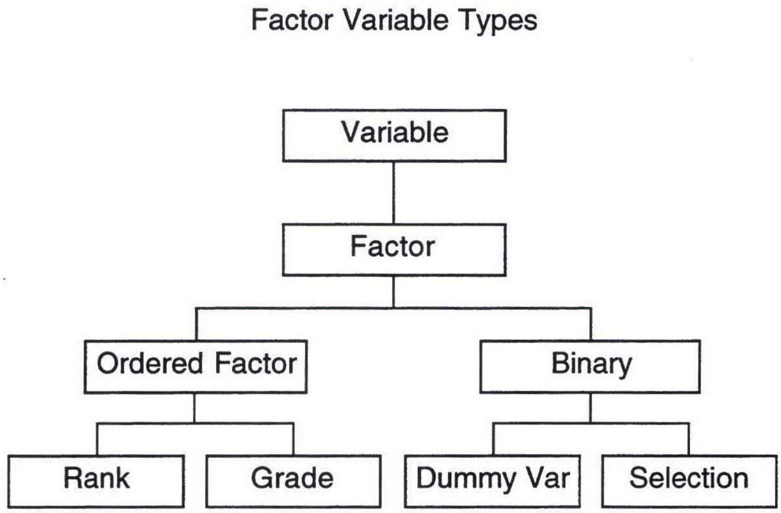


Figure 4. Variable Hierarchy for Factor Types

It is possible to have other types of hierarchical relationships among the variables other than type ("is-a") relationships ("a-part-of" relationships are very common in graphics programming) and to permit inheritance of meta-data along those other hierarchies (the KR object system, Giuse [1990], implements multiple inheritance relationships). For the NAEP data, the source of the data—student questionnaire, teacher questionnaire, school questionnaire, proficiency score, multiple-choice item, constructed response item—carries a lot of meta-information. For example, because of the sampling units, teacher questionnaire items are not really appropriate response variables.

4.1 Meta-data attached to variable objects

NAEP-VUE will implement variables as first-class objects, in an object system which is visible and extensible by the both the primary and secondary analyst. (Almond [1995] has implemented a similar variable object system for graphical models.) Variables will have attached *properties* which define important pieces of meta-data about the variable. Just like slots (sometimes called fields) of object in other object-oriented systems, properties will be inherited from variable which are supertypes of a given variable.

The following is the current working list of variable properties:

1. *Definition/Meaning* Plain English description of the variable.
2. *Extraction Query/Derivation Formula* In NAEP-VUE, all variables either come directly from the NAEP database or are derived from one or more other variable via a formula. In the former case, this slot would store a SQL or Perl script for extracting data from the NAEP database in the latter it would store the derivation formula. In particular, this means that the secondary analyst doesn't need to worry about extracting data from the database, NAEP-VUE figures out the code necessary to do this.
3. *Question* For survey questions, this would be the text of the item. For test composite scores, this might be a pointer to test objectives and sample questions (possibly in a hypertext manual). In some cases, it might be possible to use pointers to released items.
4. *Precision/Discreteness/Units* Precision meta-data can be used to remove distracting extra digits from tables and summaries. The units can be used to make table, graphs and other summaries clearer. Hand [1993] suggests some other uses for units as meta-data.
5. *Level of Analysis* NAEP data is gathered at various levels including state, school, class and student. NAEP-VUE must adjust (replicate or summarize) variables to the current working unit of analysis.
6. *Plausible Range* Often the theoretical range of a variable is much larger than the range commonly seen or expected in practice. For example, a count theoretically could be infinite, but in practice rarely be larger than 100. This information would be used for both eliciting appropriate values from the analyst and as a crude outlier detection technique. (These ranges were an important part of the ElToY design; Almond [1994]).
7. *Skewness/Kurtosis* Information about the shape of the distribution. In particular, these could be statistics calculated on the fly by the program. John Tukey (private communication) suggested looking at powers of 2 quantiles of the distribution (median, quartiles, eighths, sixteenths, i.e., the letter values) To assess skewness, the computer would look at the ratio of the distance from the median for those above the median and those below. To assess kurtosis, the computer would compare say the ratio of the intereighth range over the interquartile range to what would be expected in a normal distribution.
8. *Missingness* Data in the NAEP survey are missing due to both sample design (a balanced incomplete block design is used to administer test items) and nonresponse (schools and students declining to participate and items and questionnaire items omitted). NAEP uses multiple imputation (Rubin [1978]) to handle both types of missing data; the primary analysts develop multiple plausible values for each omitted value. Missingness properties indicate the existence of plausible values, link the multiple imputations and provide a link to indicator variables which show which observations were actual and which were imputed.
9. *Remarks/Notes* This is a place for free-text notes by an analyst. The primary analyst supplies the remarks, while the secondary analyst generates the notes.
10. *Advisory Daemons* These are little piece of code which are run at various stages in the analysis. Section 5.2 describes some possible times and applications. A list type of inheritance which would allow daemons to be cumulative up the inheritance tree and specifically overridden at lower levels (Myers *et al.* [1992]).

Properties can be stored at other levels as well, for example, model information, fit information, case information, or even information tied to specific values (although this entails a possibly too great overhead). On particularly important kind of information would probably rest with variable relationships. In particular, information about nesting and collinearity would exist on those relationships. Similarly, derived variables would have natural ties to those values which were used in their derivation.

4.2 Advisory Daemons

Advisors are small pieces of code which get attached to variable objects. They are run at certain stages in the analysis, i.e., certain user actions such as adding a variable to the model will cause NAEP-VUE to run all of the appropriate advisors.

The advisory daemons could be run at the following times: (1) Variable added to model or candidate area. (2) Interaction term added. (3) Variable selected as response or weights. (4) Model Fitting requested. (4) Model Visualization requested. (5) Variable Visualization requested. (6) Diagnostic Visualization requested. (7) *P*-value calculation (this can be used to produce guidance about multiple comparisons)

For example, one advisory daemon attached to the response variable selection actions could check the source of the variable. If the analyst chose a background variable from the teacher questionnaire as a response, a warning message would be added to the message queue reminding the user that the NAEP sampling units are students not teachers and that these data will not necessarily be reflective of National averages. Another advisory daemon attached to that slot might automatically set the model type appropriately for logistic regression if the response variable was binary. Another type of daemon attached to a variable would check for nesting constraints and when a variable is added to the model which is nested in another, it would automatically add as nested.

5.0 Interaction and Messages

Data analysis is seldom a linear process. Cowley and Whiting [1986] describe it as a tree shaped process which unfolds over time. NAEP-VUE will support this non-linear analysis of data with two mechanisms: an message queue listing suggestions offered by the advisory daemons, and a history browser.

5.1 Messages

The advisory daemons must be able to post recommendations to the user, but many pieces of advise will be difficult to manage. An agenda of recommendations which NAEP-VUE has posted, like the one used in the DETENTE system (Wroblewski *et al.*[1991]), could help the analyst manage that information. In DETENTE, the items on the agenda are *recommendations* (indicating that they have the force of "might" not "must") to the user. These recommendations are in turn supported by *task resources* which might be pieces of code or text which support the task operation. In NAEP-VUE, pointers into the NAEP documentation, on-line statistical texts, or references to off-line statistical texts would also be potential task resources.

Messages to the user would have three different levels of severity (currently coded by traffic light color):

Red *Error* — Needs User intervention/acknowledgement. Either errors in the computation or places where the user must clarify intentions.

Yellow *Warning/Advise* — Possible User Choice point. Limitations on the analysis which should be understood and places where the system has or analyst has made a default choice and the system recommends exploring alternatives.

Green *Information* — No user action recommended. Place where the computer wants to give information without a recommendation for action.

5.2 Comparing points in history

Many operation in data analysis involve comparing the analysis from different models. To facilitate this GRAPHICAL-BELIEF (Almond [1995]) introduced the notion of a history probes which compares the value of a statistic at two different stages of the model construction/manipulation process. One part of GRAPHICAL-BELIEF was a *history browser* which allows the analyst to select two states in history to compare. For the NAEP-VUE system, the history would need to be tree shaped, allowing the kinds of interactions explored in the DEXPERT system (Lorenzen *et al.* [1993]).

6.0 Future work

The ideas described in this paper are still designs, the implementation is part of an ongoing research project. Many of these designs are based on ideas prototyped in GRAPHICAL-BELIEF (Almond [1995]) so they should be within the reach of current technology. Any suggestions would be greatly appreciated.

This work was supported by a grant from the National Center for Education Statistics, award number R999B60011.

References

Anglin DG and Oldford RW [1994]. "Modelling Response models in Software." in Cheeseman P and Oldford RW (eds.) *Selecting Models from Data: Artificial Intelligence and Statistics IV*, Springer-Verlag, 413–424.

Almond, R.G. [1994]. "ElToY: Implementing Bayesian Computation Through Constraints." StatSci Research Report 24. (Under Revision for Publication)

Almond, RG [1995]. GRAPHICAL-BELIEF *Overview*, World Wide Web Site, `http://bayes.stat.washington.edu/almond/gb/graphical-belief.html`

Almond, R.G. and Schimert, J. [1995]. "Missing Data Models as Meta-Data." StatSci Research Report 29. Presented at the *5th International Workshop on AI and Statistics*, Ft. Lauderdale Florida.

Chambers, John M. and Hastie, Trevor J. [1992]. *Statistical Models in* S. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Cleveland WS [1993]. *Visualizing Data.* Hobart Press.

Cowley, P.J. and Whiting, M.A. [1986]. "Managing Data Analysis through Save-States." *Computer Science and Statistics: 17th Symposium on the Interface* Allen, D.M. (ed.), Elservier Science Publishers, 121–127.

Faraway JJ [1992]. "On the Cost of Data Analysis." *Journal of Computational and Graphical Statistics,* **(1)**, 213–230.

Giuse, D.A. [1990]. "Efficient Knowledge Representation Systems." *Knowledge Enginnering Review.* **5**(1), 35–50.

Gruber TR [1991]. "The role of common ontology in achieving sharable, reusable knowledge bases." In Allen JA, Fikes R, and Sandewall E (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference,* Morgan Kaufmann, 601–602.

Hand DJ [1994]. "Statistical Strategy: Step 1." Cheeseman P and Oldford RW (eds.) *Selecting Models from Data: Artificial Intelligence and Statistics IV*, Springer-Verlag, 3–11.

Hand DJ [1993]. "Measurement scales as metadata", in Hand DJ (ed.) *Artificial Intelligence Frontiers in Statistics: AI and Statistics III.* Chapman and Hall, 54–64.

Jacobi D and Stevens JJ [1996]. "Teacher's Professional Preparation and Classroom Instructional Practice in Relation to NAEP Mathematics Achievement" Paper presented at AERA conference, New York.

Madigan, D. and A.E. Raftery [1994]. "Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window." *JASA*, **89**, 1535–1546.

Mosteller, F. and J.W. Tukey [1977]. *Data Analysis and Regression,* Addison-Wesley, Reading, Massachusetts.

Myers, B.A., D.A. Giuse, R.B. Dannenberg, B. Vander Zanden, P. Marchal, E. Pevin, A. Mickish, J.A. Landay, R. McDaniel, and V. Gupta [1992]. "Garnet: The Garnet Reference Manuals, Revised for Version 2.0" Carnegie Mellon, School of Computer Science, Technical Report, CMU-CS-90-117-R2.

Lorenzen TJ, Truss LT, Spangler WS, Corpus WT, and Parker AB [1993]. "DEXPERT: an expert system for the design of experiments." in Hand DJ (ed.) *Artificial Intelligence Frontiers in Statistics: AI and Statistics III.* Chapman and Hall, 3–16.

O'Reilly, PF, Zelenak, CA, Rogers, AM and Kline, DL [1996]. "National Assessment of Educational Progress 1994 Trial State Assessment Program in Reading Secondary-Use Data Files User Guide". National Center for Education Statistics publication.

Raudenbush, S. [1988]. "Educational applications of hierarchical linear models." *Journal of Educational Statistics,* **13** (2), 85-116.

Riddle P, Fresnedo R, and Newman D [1995]. "Framework for a Generic Knowledge Discovery Toolkit." In D. Fisher and H-J Lenz (eds.) *Learning from Data: AI and Statistics IV* Springer Verlag, New York, 343–352.

Rogers, AM [1995]. "NAEPEX: NAEP Data Extraction Program User Guide." Princeton, NJ: Educational Testing Service.

Roth SF, Kolojejchick J, Mattis J and Goldstien J [1994]. "Interactive Graphic Design Using Automatic Presentation Knowledge." In *Human Factors in Computing Systems: CHI '94 Conference proceedings,* ACM Press, 112–117.

Rubin DB [1978]. "Multiple imputation in sample surveys: A pheonomenological Bayesian approach to nonresponse." *Proceedings of the Survey Research Methods Section,* American Statistical Association, pp 20-44.

Thoma HM and Goodall C [1991]. "Graphical Models and Their Representation." In Keramidas EM (ed.) *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface.* 30–37.

Whittaker J [1990]. *Graphical Models in Applied Multivariate Statistics,* Wiley.

Wilkinson GN and Rogers C E [1973]., "Symbolic Description of Factorial Models for Analysis of Variance." *Applied Statistics,* **22,** 392–399.

Wroblewski, David A., McCandless, Timothy P. and Hill, William C. [1991]. "DETENTE: Practical Support for Practical Action" in *Human Factors in Computing Systems: CHI'91 Conference Proceedings.* ACM Press.