An Application of Model-Based Reasoning in Experiment Design

Andrew B. Parker,[1] Sun Microsystems, and
W. Scott Spangler, General Motors

Scientists and engineers in diverse fields such as manufacturing, medicine, and design use experiments to learn about processes and the behavior of systems. Experiments study how the settings of a series of factors affect one or more response variables. For example, an engineer trying to develop a reliable painting process for automobile components might set up an experiment to study how paint viscosity and temperature (two factors) affect a numeric measure of paint surface quality (a response variable). Because an experiment can require significant resources, the experimenter often must make trade-offs between the number of experimental trials, the order of these trials, and the expected amount and type of information gained as a result of running the experiment. *Design of experiments* is the field of statistics that addresses the problem of creating layouts (ordered lists of factor combinations, or trials, to be tested) that will provide the experimenter with statistically sound results yet account for the constraints under which the experiment must be run.

The use of experimentation in industry has grown as companies attempt to reduce manufacturing costs, explore innovative manufacturing techniques, and improve product quality and safety. In practice, however, some experimenters do not use an appropriate experiment design or use no formal design at all. Often they choose a standard design from a textbook that might not be well suited to their experimental circumstances. At best, these designs can waste experiment resources because they haven't been tuned to the experimenter's goals and interests. At worst, the experimenter might not be able to draw meaningful conclusions from the experiment or might draw incorrect conclusions.

Expert statisticians can design highly tuned experiments that focus on the specific factors or interactions that are of interest to the experimenter but keep costs to a minimum. The design process is a complicated one that often requires extensive calculations and custom computer programs. After the experimental data are collected, statisticians perform a variety of complicated analyses that also require custom programming. Experimenters in industry generally lack the statistical background to produce these designs themselves and can also lack access to statisticians who can do it for them.

In response to the growing demand for statistical expertise, statisticians at General Motors Research (GMR) developed a new approach to experiment design based on a unified mathematical model (Lorenzen and Truss 1990). The approach allows experimenters to apply a theoretically sound design methodology to satisfy a number of experimental goals and conditions. Although this methodology was successfully taught to both statisticians and nonstatisticians, it still proved time consuming and required extensive training for nonstatisticians. They decided to automate the process by building an expert system for design of experiments and the analysis of experimental results. The goal was to allow experimenters with minimal statistical expertise to produce custom, high-quality experiment designs quickly, without the assistance of expert statisticians.

Statisticians at GMR first attempted to automate their design methodology by developing C-language application programs. Although they achieved some limited success in automating some of the more tedious calculations, they soon realized that the exploratory nature of the design process would be difficult to capture in traditional computer programs. They began working with GM's Advanced Engineering Staff, who proposed an AI-based solution to be developed in conjunction with IntelliCorp, Inc., and Electronic Data Systems.

The resulting system has become known within GM as DEXPERT. DEXPERT aids an experimenter in the search for an optimal or near-optimal design. It provides manual and automated search through an infinite design space and allows the user to pursue several different design trade-offs simultaneously. It contains knowledge of numerous statistical techniques and the ability to perform all required calculations automatically. It includes graphic displays to aid the user in visualizing the design process; it also contains online help and tutorial facilities that are tailored to the user's level of expertise.

Other work in this area includes several commercially available systems for design of experiments, including CATALYST / RPE and RS / DISCOVER. Some research has been done in developing expert systems for Taguchi-type experiments (Lee, Phadke, and Keny 1989). Many of these programs simply assist users in selecting from a fixed set of standard designs. None provides the user with facilities for searching a design space or provides the rich set of design tools available in DEXPERT. Although some of these programs do use AI techniques, they use simple rule-based approaches for selecting among a fixed set of designs.

The authors attribute the successful development of DEXPERT, at least in part, to the use of model-based reasoning (Kunz 1988) in its development. *Model-based reasoning* is a development methodology that involves developing a symbolic, structural model of the application domain. The model can then be combined with a number of reasoning mechanisms and interfaces to produce several applications of the model. Model-based reasoning has allowed DEXPERT to more closely mimic the exploratory design approach used by expert statisticians. As a result, DEXPERT is complete enough to provide expert statisticians with tools that increase their productivity and improve the quality of their designs. DEXPERT provides guidance to nonstatistician experimenters to allow them to create designs comparable to those of an expert.

Experiment Design Processes

This section highlights the experiment design processes used when designing experiments manually and when using DEXPERT. Both processes use the previously cited design methodology developed at GMR. For the purposes of this chapter, the statistical content has been minimized and simplified where possible. Statistics concepts that are mentioned but not explained are included for readers with more statistical sophistication; they are not essential to the comprehension of the overall content of the chapter.

The following example illustrates the use of the mathematical model on which the GMR design approach is based: An automotive engineer wants to determine what causes variations in fuel economy in a certain type of car. He / she speculates that some of the effect might be caused by minor variations in the physical properties of the engine between one car and another. Another possibility is that the type of fuel injector used in the car might have an effect. Still a third factor might be the type of fuel used. There is also the possibility that particular combinations of two or three of these factors might be affecting fuel consumption. This last possibility is what statisticians refer to as an interaction between factors.

The mathematical model for this experiment would be the following:

$$Y = E[\iota] + I[\varphi] + F[\kappa] + EI[\iota,\varphi] + EF[\iota,\kappa] + IF[\varphi,\kappa] + EIF[\iota,\varphi,\kappa] + error[\iota,\varphi,\kappa,\lambda] \ .$$

The variable Y represents the fuel consumption for a particular trial of the experiment and is referred to as the *response variable*. The term $E[\iota]$ represents any effect on fuel consumption because of changing engines. The ι subscript ranges over the different engines used in the experiment (that is, if we use four engines numbered 1 to 4, then ι would be an integer from 1 to 4.). The EI term represents the effect of the interaction between engines and injectors (separate from the effect of changing engine or injector alone). Its subscripts $[\iota,\varphi]$ range over all combinations of the different engines and injectors used in the experiment. The *error term* represents all effects on the response variable not caused by changes in engines, injectors, or fuels. The subscripts of the error term include λ, which takes on a different value each time an $[\iota,\varphi,\kappa]$ combination is repeated. Thus, the value of the error term is expected to vary for each trial of the experiment. The error term represents the underlying variation when all experimental factors are held constant.

Manual Experiment Design

When statisticians design experiments manually, they evaluate various properties and computed characteristics of each term to determine the conclusions that could be drawn if specific settings and parameters were applied. In the example, the factor engines (E) have a variety of attributes based on the nature of the area of study. For example, it is qualitative (as opposed to quantitative factors such as temperature). This attribute cannot generally be changed by the experimenter but influences the statistical power of the experimental design. Other attributes can be manipulated by the experimenter. For example, the experimenter can manipulate the number of levels of a factor in the experiment to influence the experimental design's power. In the example, the factor engines will be studied at four levels; that is, four different engines will be tried in the experiment. The experimenter can decide to test an additional engine or choose to eliminate one of the original four from the study, if necessary, to reduce the total number of trials to be done.

In addition to the qualitative-quantitative distinction and the number of levels, a variety of other characteristics must be determined for each factor. The statistician notes these characteristics and uses them to determine a series of measures for each term in the mathematical model. The statistician then uses the mathematical model and term information to compute, for each term in the model, an equation that describes the experiment's ability to test the effect or interaction represented by that term. For a particular set of experimental circumstances (for example, each of the factors engines, injectors, and fuel studied at two levels, without repeating any combinations of trials), the statistician can compute that the experiment will detect medium-sized differences in fuel consumption caused by a change in fuel type but might not be able to determine if the interaction between engine, injector, and fuel type has any effect on fuel consumption. Computing these expected mean squares and detectable differences before running an experiment allows the statistician to evaluate the potential costs and benefits of a particular design before actually running the experiment.

Each permutation of the experimental circumstances requires the statistician to perform complex symbolic manipulation of equations, calculations, and bookkeeping for comparison between designs. These calculations can be complex, especially taking into account more advanced statistical concepts such as nesting and fractional designs. In practice, experienced statisticians don't have the time to calculate and analyze more than one or two sets of experimental circumstances for a given experiment, especially when the experiment involves several factors or complex relationships between factors.

After a design is selected, the experimenter must generate a randomized layout sheet that will be used in data collection. After the experimental data are collected, the experimenter performs a series of statistical analyses. Often, these analyses are performed repeatedly, after making assumptions or transformations based on the initial results. Finally, the experimenter creates charts and graphs, accompanied by descriptive text, to illustrate the experimental results. Each of these steps often requires the experimenter to write several custom computer programs using statistical software packages such as SAS. The design and analysis process might take a highly trained experimenter anywhere from several days to several weeks, not including the time required to run the experimental trials.

Experiment Design Using DEXPERT

When designing experiments using DEXPERT, the experimenter interacts with a window-oriented user interface. The interface provides assis-

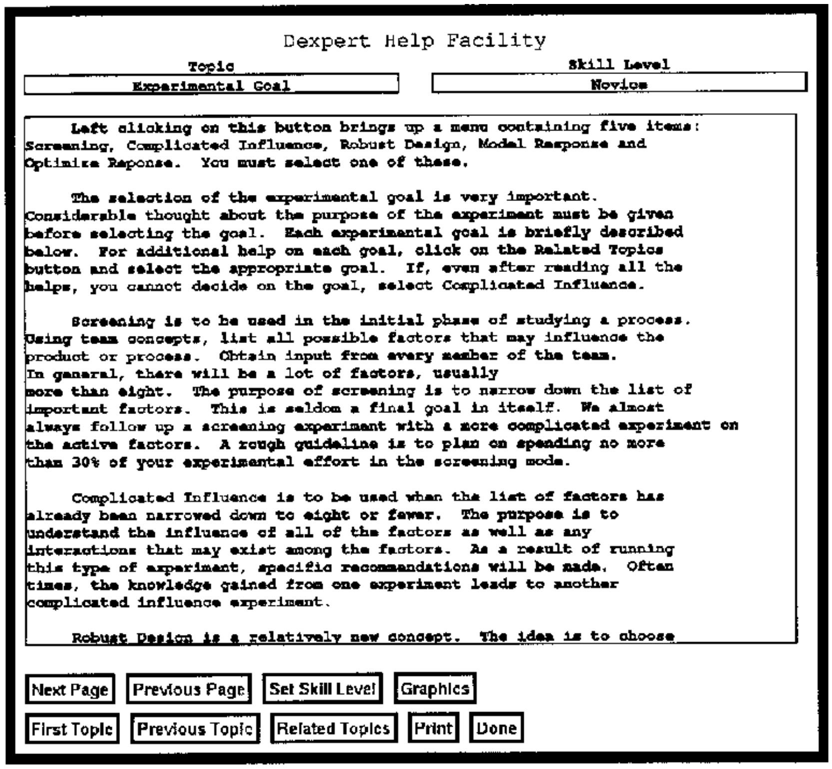

Figure 1. An Example Display from DEXPERT*'s Help Facility.*

The user obtains help by right clicking with the mouse on fields, buttons, or menus in DEXPERT *displays.*

tance to the user based on the self-described level of statistical expertise: novice, intermediate, or expert. The instructions, definitions, and explanations received are tailored according to this expertise level. The user begins the DEXPERT session by specifying the high-level experimental goals as well as information about the experimental factors and response variables. DEXPERT checks for inconsistencies and assists novice users in specifying various attributes, for example, whether a factor is quantitative or qualitative. At any time, online help is available to assist the user in answering questions or understanding the displayed information (figure 1). The display shown in figure 2 summarizes the initial information entered by the user.

Based on the initial information, DEXPERT generates an initial design and computes the statistical power for each term in the mathematical model. The relevant features of the initial design are presented in a tabular format that is tailored to the profile of the user, as shown in

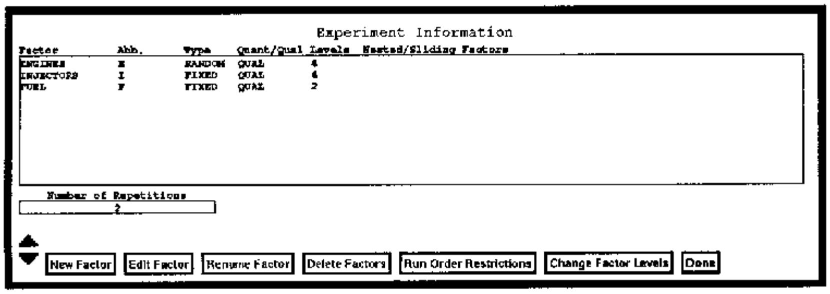

Figure 2. A Display Summarizing the Information That an Experimenter Has Specified about the Experimental Goal and Factors in the Experiment.
Buttons along the bottom of the display allow the user to modify the factor information before proceeding to the design phase.

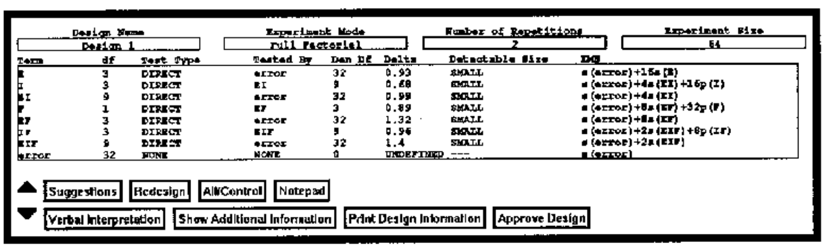

Figure 3. The Initial Design Display Based on the Experimenter's Initial Specifications.

The columns that appear in the table vary based on the user's preferences and level of expertise. This example display shows all available information. Buttons along the bottom of the display allow the user to annotate the design, request suggestions or textual interpretations, or redesign using a variety of techniques.

figure 3. The user can then modify the design input (that is, redesign) to achieve the desired experimental characteristics. In addition, the user can request expert advice from DEXPERT on how to modify the design to achieve desired statistical properties. Based on the user's input about the aspects of the design needing improvement, DEXPERT generates suggestions (figure 4), which the user can apply to create a new design (figure 5). The user can also request a textual critique of the statistical power of a design (figure 6) or a comparison of two designs.

DEXPERT provides a complete set of redesign options at each design stage: The user can fractionate a design in many different ways and can specify such properties as maximum design size, terms of interest, terms not to be confounded with terms of interest, resolution, and

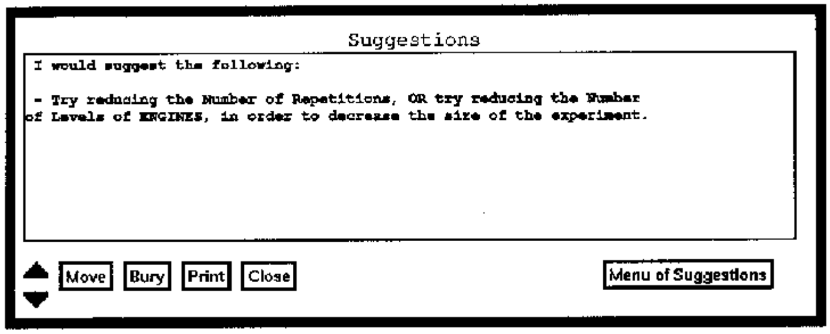

Figure 4. A Display Showing Redesign Suggestions Generated by DEXPERT *in Response to the Experimenter's Critique of the Power or Practicality of the Current Design.*

The menu of suggestions button allows the user to automatically pursue one or more of these suggestions to generate new designs.

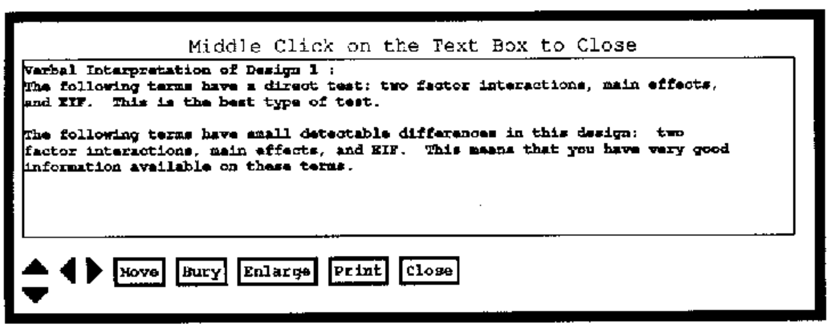

Figure 5. A Display Showing a Second Design Created by Reducing the Number of Repetitions from the Initial Design from Two to One.

In this display, the user reduced the columns of information displayed to those typically displayed for a novice user.

blocks of fractionated factors. DEXPERT allows the user to request restrictions on randomization because of experimental constraints and calculates the loss of information that results from the restriction. The user can specify that some terms or groups of terms can be assumed negligible for the purposes of the experiment. Many of these options require DEXPERT to perform substantial calculations and generate custom SAS programs—a time-consuming and often impractical task when done manually.

The experiment design process using DEXPERT allows for a much

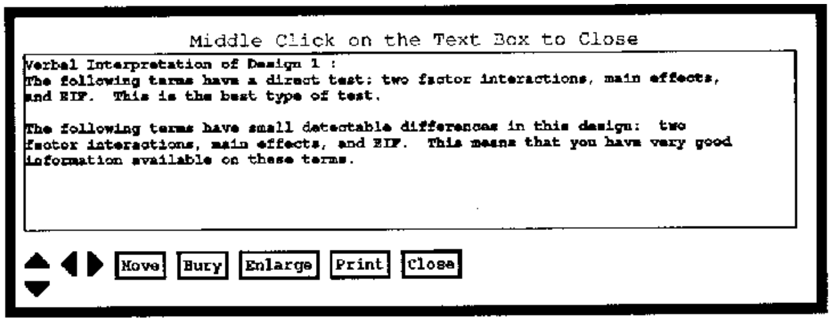

Figure 6. A Display Showing DEXPERT's Textual Description of the Initial Design.

more thorough evaluation of potential experiment design alternatives. DEXPERT computes the statistical power of each design quickly. It allows for the generation of redesign alternatives from the initial design and all succeeding designs, resulting in a hierarchy of designs (figure 7). The user can also specify the desired detectable differences for each term in the mathematical model and have the system automatically search for the design with the smallest number of trials that meets the criteria. The user iterates the redesign process, often creating dozens of possible designs, until one or several satisfactory designs is generated. The user then selects a design, and DEXPERT generates the randomized layout sheet for data collection.

Once the data are collected, the experimenter returns to DEXPERT and enters it into the system interactively or through a text file. DEXPERT analyzes and interprets the results using a number of analytic and graphic methods. Figure 8 shows a standard analysis of variance table display in DEXPERT. Analysis is performed separately on each response variable. DEXPERT provides a textual interpretation of the statistical analysis on request, as shown in figure 9. It provides a number of additional analysis features such as data transformations, regression analysis, predictions of response values for particular factor-level combinations, and comparison of means. A variety of graphic analyses are provided, including main-effect plots of means (with confidence intervals), interaction plots, effects plots, residual plots, time plots, response surface plots, and contour plots, all of which can be printed as well as appear on the screen.

In contrast to the manual approach, even an inexperienced experimenter can generate a design comparable to that of an expert in 30 to 90 minutes. After collecting experimental data, the experimenter can generally complete the statistical analyses and graphs within a few hours.

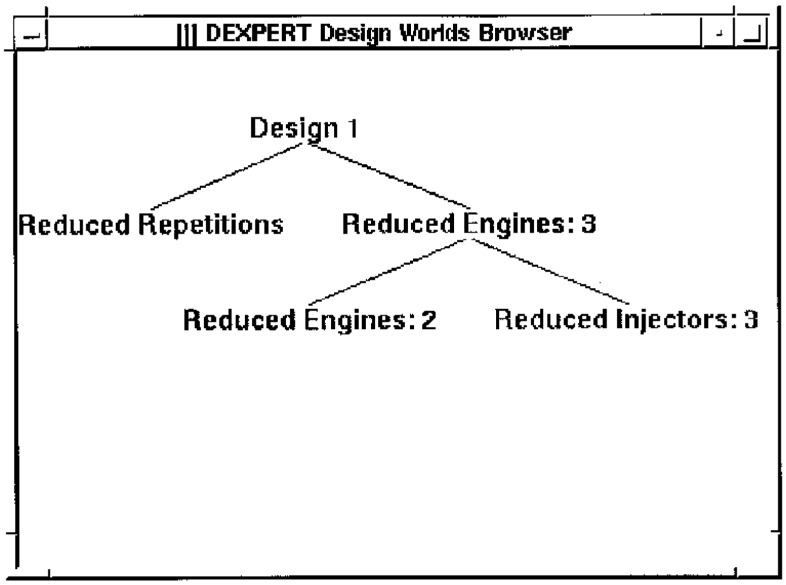

Figure 7. A Display Showing the Hierarchy of Designs Created during a Design Session.

The user can click on a design name to display or rename the design.

DEXPERT performs all calculations, generates and submits all required SAS programs, and produces a variety of printed reports and graphics.

Model-Based Reasoning in DEXPERT

A symbolic model of design of experiments theory forms the core of DEXPERT. The symbolic model represents the mathematical model that describes the experiment as well as behaviors of, and relationships between, components of the model. The reasoning components of the system manipulate the model's attributes and relationships using techniques similar to those of an expert statistician using paper-and-pencil models with calculators or statistical software packages. DEXPERT also uses the model to assist in performing analyses of the experimental results after the experimenter has collected data. The model-based approach also facilitated an interactive development strategy that allowed the system to be used early in the development cycle while more complex reasoning techniques were being added.

An object-oriented frame system is used to implement the symbolic model. Hypothetical reasoning, rule-based reasoning, and state-space

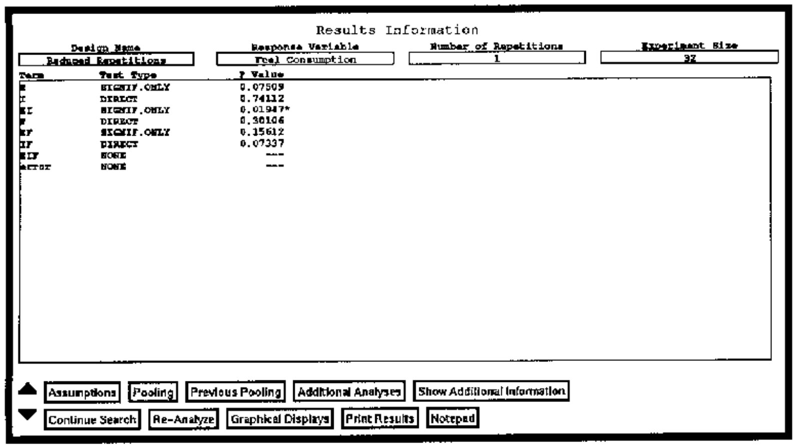

Figure 8. DEXPERT*'s Display of the Analysis of Variance Table for the Experimental Results.*

As with the design display, columns that are not appropriate to the user's profile were omitted. A variety of additional analyses and graphic displays are available to the user through buttons along the bottom of the display.

search are utilized to implement the statistical design strategies. A variety of different user interface components interact with the model through procedural code. The reasoning and procedural components of the system interact with external routines, including C-language programs and dynamically generated SAS programs to perform some of the more difficult calculations.

Object-Oriented Frame Representation

The symbolic model is constructed using the KEE object-oriented frame system (Fikes and Kehler 1985) to represent the factors in the experiment, the terms in the mathematical model, the user's goals and level of statistical expertise, and the relationships between these components. Once constructed, this model is manipulated and evaluated using a variety of reasoning tools and procedural techniques. In addition, the user can manipulate the model or provide guidance to the system through the user interface.

For example, DEXPERT creates a frame corresponding to each factor to be studied in the user's experiment. Attributes of each factor are stored in the slots on the factor frame. In the previous example, the factor engine would have the following slot values filled in:

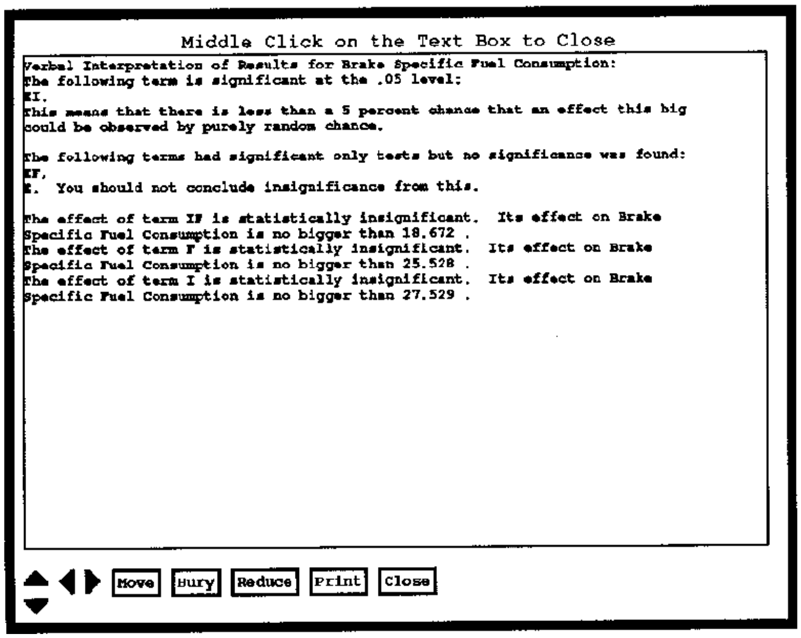

Figure 9. A Display Showing DEXPERT*'s Textual Explanation of the Results for the Experiment.*

factor.abbreviation:	E
quantitative.or.qualitative:	QUALITATIVE
fixed.or.random:	RANDOM
number.of.levels:	4

...

DEXPERT also represents the terms in the mathematical model as frames. Attributes for each term are stored in the slots on the term frame. In addition to computed attributes, the model also maintains structural relationships between terms. For example, the tested.by relation links terms that provide statistical tests on other terms.

Hypothetical Reasoning in Multiple Worlds

One of DEXPERT's greatest strengths is its ability to search and evaluate many alternative designs. While the user searches for a design best suited to his/her needs, DEXPERT keeps the alternate designs available for comparing and for returning to previous design states. The challenge for DEXPERT was to provide many alternative designs in parallel (that is, without backtracking) to allow the user to participate in the exploration process. The complexity of the symbolic model (often consisting of hundreds of frames and thousands of attributes) made it impractical to duplicate the symbolic model for each design iteration. Instead, a context or *worlds* mechanism (Filman 1988) was utilized to represent each design as a hypothetical state of a single symbolic model. The symbolic model exists in the background, an initial state containing the frame hierarchies, slots, relationships, and methods for the user's experiment. Each design is represented as a world, offering a hypothetical configuration of this background model. World-dependent facts include relationships between factors and sets of terms in the mathematical model as well as numerous user-specified and system-computed attributes of factors or terms. The design worlds are arranged into a hierarchy that preserves the history of the design session and allows the state of the model to be inherited between parent and child worlds. As a result, only changes from a previous design need to be recorded in each new design.

During the design session, the user explores new hypothetical designs by asking the system to apply one of the redesign options to the current hypothetical world. Each such action adds a new design world to the hierarchy of designs, modifies the relationships among components within that design world, and updates any computed attributes that apply to the new model structure. The user iterates this process until satisfied that the best possible design was found. Through this mechanism, an inexperienced user can often generate and examine dozens of alternative designs during a session, where an expert statistician using manual methods might spend days investigating a small subset of these designs.

Using worlds in conjunction with the underlying model also proved valuable in implementing the analysis components. In the initial analyses, DEXPERT calculates a variety of statistical measures based on the collected data and the information stored in each term frame. Based on the initial results, the user can apply alternative analyses that might be appropriate for the user's experimental goal, or the user can refine the analysis by applying various transformations to the standard statistical techniques. Each of these alternative analyses is created as a child world of the initial analysis, inheriting the required data but storing the results of the specialized analysis locally.

Rule-Based Reasoning

DEXPERT utilizes production rules to complement the knowledge represented in the symbolic model and the state of the worlds system. The rules are partitioned into rule classes that can be invoked independently. When the user requests assistance in redesigning the experiment, the system invokes one or more rule classes to evaluate the state of the model in the current design world. The rules produce a series of redesign suggestions that are presented to the user. After acceptance by the user, these suggestions can then be applied to the current design to create a new design. The classification of rules into rule classes allowed the system to apply specialized expertise for different experiment goals as well as provide for prioritized redesign techniques.

Each rule in the knowledge base concludes with both a suggestion and a reason for the suggestion. The rule base was set up this way to address the problem of conflicting suggestions. For example, if a user wants to decrease the size of the current design while he/she improves the information available, then it would make sense to decrease the levels of engines to satisfy one goal (smaller experiment size) and increase the number of repetitions to satisfy the other goal (better information). If the system simply made these two suggestions at the same time, it might appear contradictory to the user. Relating suggestions to the user's goals provides a better understanding of the trade-offs involved in trying to achieve an experiment design that meets all the user's requirements.

Automated State-Space Search

Hypothetical worlds used in conjunction with the symbolic model facilitate DEXPERT's application of search techniques to assist the user in finding an optimal design. The design worlds correspond to nodes in a search tree that can be evaluated based on figures calculated from the state of the model. Arcs in the search tree represent the application of a subset of redesign options permissible in the user's experimental domain. When requested by the user, DEXPERT will perform a search using a search algorithm know as *branch and bound with dynamic programming* (Winston 1984). The algorithm finds the smallest design (in terms of number of trials) having the user-specified set of detectable differences by sorting the search queue in order of increasing design size. The user first specifies the attributes that are practical to vary in the context of his/her experiment. New designs are generated by incrementally changing each specified feature in the smallest design in the queue. The search terminates when one or more satisfactory designs are found, or the design size exceeds a maximum that is controlled by the user. The automated search facility is an example of how the underlying model made it easy to add another tool (a search algorithm) to the system to provide a valuable capability to the experimenter.

Textual Interpretation

DEXPERT's model-based representation of the design closely mimics conceptual and statistical models of the experiment. This representation facilitated the development of features in DEXPERT that translate the internal representations of the design conclusions into English summaries that are easy for the user to understand. By sorting the term frames in a design by their test type and their minimal detectable difference categories, DEXPERT generates a concise textual description of the properties of a particular design. A similar strategy is used to produce text that compares two different designs. DEXPERT's explanations of the experimental analysis results also demonstrate this technique. In each case, attributes and relationships inherent in the underlying model are interrogated by procedural code to produce the explanations. These features demonstrate the power of the model-based approach to add applications of the system's inherent knowledge. Although they were simple to implement, these features provide valuable assistance to the user that other experiment design packages lack.

Application History and Benefits

Development on DEXPERT began in June 1989. By January 1990, the statisticians at GMR were using preliminary versions of DEXPERT to design real experiments. A small number of engineers began prepilot use of DEXPERT in late 1990. Pilot release of the complete DEXPERT system began in January 1991, followed by a full production release in May 1991. As of this writing, over 60 sites within GM use DEXPERT on a regular basis, averaging once each week. DEXPERT has been used, primarily by engineers, to design and analyze hundreds of experiments within GM.

DEXPERT was first deployed to a small number of pilot sites that received individualized training. A DEXPERT consulting center was established, including several workstations available for general use by engineers. A short training course was developed and conducted at the consulting center; to date, over 250 engineers and statisticians from GM have completed the DEXPERT training.

Full deployment of DEXPERT required porting the system to optimized, run-time versions of the Lisp and KEE system software and subsequent ports from SUN workstations to UNIX workstations from IBM and Hewlett Packard. None of these activities proved notably difficult. It should be noted that the KEE worlds facility used to implement the design worlds was replaced by a simpler, custom-coded facility to improve run-time performance. Deployment activities also included identifying and educating the potential user community within GM, establishing an informational newsletter and support hotline for users, and coordinating the installation of hardware and software at user sites.

An estimated eight person-years of effort went into DEXPERT's development, and an additional four person-years of effort have gone into the initial deployment and maintenance efforts. The system will be maintained in the future by less experienced maintenance staff members, with assistance from the expert statisticians at GMR. The domain knowledge is expected to remain relatively static over time, with maintenance activity primarily associated with bug fixes, ports to new platforms, and minor functional improvements as requested by users.

The statisticians who used DEXPERT in early consultations with engineers kept careful records of estimated experimentation cost savings that resulted from each consulting session. During pilot testing, engineers who came to the DEXPERT consulting center to use DEXPERT were asked to quantify the savings owed to efficiencies gained from using DEXPERT. It was not uncommon for a DEXPERT session to result in cost savings of over $100,000 for a single experiment. The long-term benefits to GM associated with the use of DEXPERT are estimated to be in the millions. In addition to savings in experimentation costs, these benefits include reduced costs because of the development of more efficient manufacturing processes, lower warranty costs, and increased revenues because of improvements in product quality.

Conclusions

Although many expert systems focus on single-paradigm approaches to encoding knowledge (such as production rules), experts rarely use a single technique or type of knowledge in solving a problem. More often, an expert will draw on a large body of background knowledge that constitutes a model of the application domain and apply a variety of techniques until a satisfactory solution is achieved. DEXPERT closely approximates a design approach used by expert statisticians through its hybrid architecture. The architecture combines a symbolic model of the domain area with hypothetical worlds, production rules, search algorithms, and object-oriented programming. Because DEXPERT represents a complex model and state space in a computer program, it can manipulate the model much faster than a human expert and allows the expert to generate improved designs by exploring many more design alternatives in less time. A window-oriented interface customized for different classes of users allows experimenters with minimal statistical training to generate designs that are comparable to those of an expert.

The inclusion of analysis techniques with the design capabilities allows the novice experimenter to conduct an entire study without statistical consultation or the writing of custom computer programs.

DEXPERT was developed and delivered to users in an iterative manner. Statisticians were using early versions of DEXPERT for real experiments even as it was being developed. The model-based approach made it possible to provide a core model of the fundamental concepts required for the system. Once this model was put in place, it was easy to later add new reasoning techniques and interfaces to the existing model with minimum effort.

In January 1992, the inventors of DEXPERT were honored with the Kettering Award, GM's highest technical honor. This award recognizes the benefits to GM that have been and continue to be realized by DEXPERT users throughout the company. In less than one year since its initial deployment, DEXPERT has become the standard tool for experiment design at GM.

Acknowledgments

The authors would like to thank Tom Lorenzen and Lynn Truss at General Motors Research, who developed and put into practice the statistical theory on which DEXPERT was built. It was through their vision that DEXPERT was created; they were instrumental in all phases of DEXPERT's development. Bill Corpus of Electronic Data Systems wrote substantial portions of DEXPERT. Catherine Perman worked on the design of DEXPERT in its early stages. The authors would also like to thank the following reviewers for their thoughtful comments on earlier drafts of this chapter: Laura Paterson and Tony Confrey; Robert Filman, Christopher James, and James Taylor at IntelliCorp; Bret Fisher and Mitchell Levy at Sun Microsystems; and Karon Barber at General Motors.

Note

1. Andrew B. Parker was previously a senior knowledge systems engineer at IntelliCorp.

References

Fikes, R., and Kehler, T. 1985. The Role of Frame-Based Representation in Reasoning. *Communications of the ACM* 28(9): 904–920.

Filman, R. E. 1988. Reasoning with Worlds and Truth Maintenance in a Knowledge-Based Programming Environment. *Communications of the ACM* 31(4): 382–401.

Kunz, J. C. 1988. Model-Based Reasoning in CIM. In *Intelligent Manu-*

facturing: Expert Systems and the Leading Edge in Production Planning and Control. Reading, Mass.: Addison Wesley.

Lee, N. S.; Phadke, M. S.; and Keny, R. 1989. An Expert System for Experimental Design in Off-Line Quality Control. *Expert Systems* 6(4): 238–249.

Lorenzen, T. J., and Truss, L. T. 1990. Anatomy of DEXPERT—An Expert System for the Design of Experiments, Research Publication GMR-7111, General Motors Research Laboratories, Warren, Michigan.

Winston, P. H. 1984. *Artificial Intelligence*. Reading, Mass.: Addison Wesley.