AI Magazine Volume 7 Number 5 (1986) (© AAAI)

John Kastner, Chidanand Apté, James Griesmer, Se June Hong, Maurice Karnaugh, Eric Mays, and Yoshio Tozawa

A Knowledge-Based Consultant for Financial Marketing

Introduction

This article describes the initial stages of an effort to develop a knowledge-based financial marketing consultant system. The project for Financial Marketing Expertise (FAME), is to produce a system that addresses the area usually referred to as *financial marketing*. This term characterizes the financial decision processes used in the marketing of products and services of such large scale that they can significantly impact a company's financial status. In particular, our project emphasizes financial marketing as it applies to the marketing of computers. For instance, a customer interested in buying computing technology on a large scale is usually concerned that the financing plan being used to acquire the technology is safe, sound, and attractive from a financial investment point of view. Therefore, in making very large sales, financial considerations often become as important as the computing considerations.

We have found financial marketing to be a very interesting and characteristically unique domain. The problem is that of generating a financing plan, and this differs from most expert systems applications, which usually tend to be classificatory in nature. Human financial marketing experts, rather than exhaustively generating an optimal plan (which might not even exist), use their experiential heuristics and domain knowledge to prune the generate-and-test space for efficiently designing a plan that is attractive from the customer's viewpoint. There have been relatively few expert systems that employ such a heuristically guided generate-and-test problem-solving paradigm for designing an acceptable plan. Systems that partially exhibit some of these facets include DENDRAL (Lindsay et al. 1980) for heuristic generate and test and MOLGEN (Stefik 1981) for plan design.

John Kastner, Chidanand Apté, James Griesmer, Se June Hong, Maurice Karnaugh, and Eric Mays are research staff members of the Expert Systems Group in the Mathematical Sciences Department at the IBM Thomas J Watson Research Center, Yorktown Heights, New York 10598 Yoshio Tozawa is a research scientist at the IBM Tokyo Research Laboratory, 5-19 Sanbancho, Chiyoda-ku, Tokyo 102, Japan

Another important distinction between this problem and many others addressed by AI researchers is that a typical financial marketing problem frequently has no one solution. There might be no definitive answer to a problem. The issue is not merely a question of computing financial optimality. The importance lies not only in the answer you provide but also in the explanation and justification that you use to back the answer. For this purpose, it is important to generate a convincing financial argument that strengthens the selling of the answer (financing plan). Explanation generation in natural language, for a solution that might be competing with many others has been addressed by very few knowledge-based systems. Some similarity exists, though, between this problem and the one faced by researchers building systems for doing legal reasoning; there, too, a problem exists in presenting a convincing argument for winning one's case (for example, the TAXMAN project) (McCarty and Sridharan 1981). However, the financial marketing domain contrasts with the legal reasoning domain in that one usually cannot make use of precedents because they are rarely available. For an automated financial marketing problem solver to generate such convincing arguments, it is crucial to accurately determine what concerns a potential customer and then to use these concerns in the plan and justification generation.

Abstract This article describes an effort to develop a knowledge-based financial marketing consultant system. Financial marketing is an excellent vehicle for both research and application in artificial intelligence (AI). This domain differs from the great majority of previous expert system domains in that there are no well-defined answers (in the traditional sense); the goal here is to obtain satisfactory arguments to support the conclusions made. A large OPS5-based system was implemented as an initial prototype We present the organization and principles underlying this system and offer our ongoing research directions. The experience gained in the initial prototyping effort is currently being used to further expert systems research and to develop an extensive system that ultimately can be used by the marketing organization.

The FAME project was begun in early 1985. The first several months were spent familiarizing ourselves with the domain, characterizing the types of problem solving in this domain, and identifying the different areas requiring expertise. While undergoing this educational process, an initial prototype system was built as a means for communicating to our potential user community the flavor of a knowledge-based approach for solving their problems. The tools used in this effort were chosen on the basis of our group's recent extensive experience in building YES/MVS (Griesmer et al. 1984; Ennis et al. 1986). The initial prototype system consists of over 700 OPS5 rules, about 40 Lisp/VM functions, and the IBM graphical data display manager (GDDM) graphics interface. The system makes full use of the color graphics capability of an IBM 3279 terminal, using a domain-independent window-management package especially developed for use by OPS5 programs.

Financial Marketing: A Vehicle for AI Application and Research

A twofold problem, which seems to require significant expertise, exists for a marketing representative: (1) the preparation of a recommendation to a customer of a capacity solution that meets the customer's computing requirements over a period of time and (2) a financial solution which outlines a plan for acquiring this capacity under financial terms and conditions that best address the customer's needs and concerns. In addition, the marketing representative must be able to justify the proposal.

For the sake of illustration, we outline a typical marketing situation—a corporation whose installed processing power will fall short of its estimated growth requirement at a certain time. The corporation management might, therefore, seek solutions to this problem from marketing representatives. A data processing executive might request proposals and forward reasonable recommendations to a financial executive who might make the final decision. The financial executive's concerns, such as the company's outlook on its earnings-per-share ratio, might differ from the data processing executive's budgetary concerns. Given this type of information, the problem is to come up with a solution that can be justified and sold to personalities with diverse criteria and concerns. Considerable expertise lies in generating a set of reasonable plans and then proposing and defending the one that is best based on the criteria and concerns of the customer.

An Example

As an example, assume that the imaginary TNW Corporation's data processing center has three IBM mainframe processors: a 3081 D16 running MVS development applications, a 3081 K32 running MVS production applications, and a 3083 JX3 running VM interactive applications.

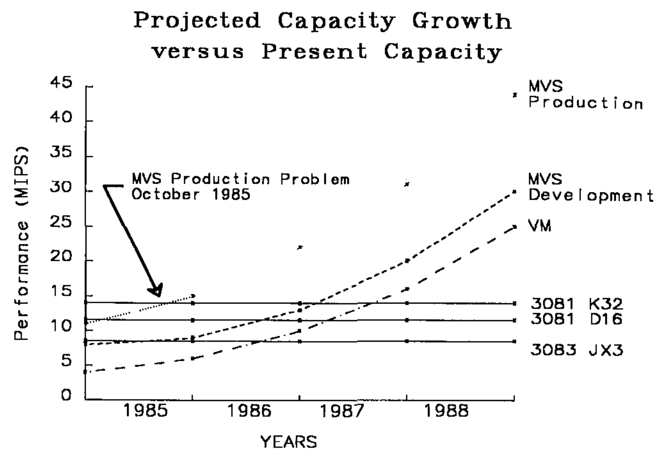

Figure 1. TNW Capacity Growth Plan.[1]

The current date is 1 July 1985. The compound growth rates in processing power requirements through the end of 1988 have been estimated to be 40% for MVS production, 40% for MVS development, and 55% for VM. Figure 1 shows these processing requirements over a four-year period relative to the current install base. The TNW vice-president in charge of information systems, having realized that his current install base will soon run out of power, has requested his marketing representative develop a capacity solution which can be implemented using an attractive financing plan. The VP's key concern is budget; he is committed to spending no more than $4 million on CPU expenditures in 1985. Although his 1986 CPU budgetary expenses have yet to be decided, the 1985 figure and the required growth in computing are fair indicators of what his future budgetary figures will be.

Also, let us assume that a competitive situation exists; a third-party leasing company by the name of ABC Leasing has proposed the following four-part solution: (1) install a used 3081 K32 from ABC Leasing on 20 October 1985; (2) remove the 3081 D16 that is currently on lease from ABC Leasing as soon as possible after 20 October; (3) replace the used 3081 K32 (installed on 20 October 1985) in June 1986 with a new 3090 200 processor leased from ABC Leasing at a good rate; and (4) move all MVS production work to the 3090 at that time (June 1986), and move MVS development to the 3081 K32 leased from XYZ Leasing when ABC's 3081 D16 is removed.

In addition to solving the computing needs problem, the marketing representative might also want to develop a financing plan which is better than that offered by the competition. In this way, the marketing representative is addressing the total financial marketing problem faced by the customer and can provide an integrated solution.

[1]Estimates of processor power are foı illustrative purposes only.

Financial marketing problems such as in the TNW case are knowledge intensive. A situation needs to be assessed using a complex collection of case-specific data points in conjunction with databases containing information on product offerings. A situation is specified by information on the customer install base, his projected growth in computing needs, his financial and organizational profile, and the marketing representative's proposed solutions (if any). The offerings database might contain information on a wide variety of subjects, ranging from specific data on processor models and options and their costs to terms and conditions of the various financing options available for acquiring such products. This large amount of situation-specific knowledge in turn requires employing significant problem-solving expertise.

Two major knowledge-intensive tasks have been identified: generating suitable capacity solutions for a customer and generating financially attractive plans for the acquisition of these capacity solutions. In general, a *capacity solution* can be defined as a series of discrete times, each associated with a set of actions (for example, upgrades, replacements, additions, and removals of processors). Each action-time pair needs to be associated with a financing method such that the overall financing plan for a capacity solution is the most attractive possible, both analytically and qualitatively. The goal is to tailor a financially attractive proposal that will solve the customer's computing growth requirement and to generate a convincing argument which will enable the "selling" of this action to the customer.

The balancing of the capacity solution with the financial-acquisition solution is one of the key issues involving expertise. It is because of these two competing goals that the financial marketing problem is not merely one of optimization. The expert in this domain must produce effective arguments for a balance of these competing goals.

Capacity Planning

Given a large customer's install base and the projected growth in processing power requirements, it is possible to generate a vast number of possible capacity solutions based on the availability of a very wide range of processors and their various models and upgrade options. Exhaustive generation of these solutions can be computationally expensive. The search space can be efficiently pruned using heuristics that retain only a moderate number of the best solutions for further financial analysis. A sample of these heuristics follows: avoid upgrading old technology, avoid upgrading processors that are nearing their lease expiration, and consider balancing processor work loads.

Acquisition Method

After arriving at a set of computing solutions to the problem, reasonable financing plans for each of the computing solutions must be determined. Here, too, a computational problem can exist. The vast number of financing options that are available, if applied to each of the computing solutions and their subsolutions (for example, if a computing solution consists of carrying out actions on three processors, then each of these actions can be performed using different financing plans), can result in a very large set of financing plans. Because these plans need to be financially analyzed to examine their impact on the tax and accounting books of the customer, it is prudent to keep the size of this plan set small. Here, too, a number of criteria based on the customer's financial and organizational profile can be used to constrain the number of financing plans generated. A sample of these heuristics follows: If the customer has any identifiable historical trends, he will probably continue following them (for example, always lease short-term solutions and purchase long-term solutions). The customer will probably prefer consistent financing plans for each of the subcomponents of a computing solution.

It is worth noting here that these heuristics also tend to keep the number of generated financing plans down to a moderate number. The major drawback of using heuristics for constraining and pruning the search space is there is no guarantee that you won't overlook a better solution for the specific problem at hand. This shortcoming can be partially compensated in an interactive system by allowing the user the ability to augment a solution set each time this technique is used.

Financial Analysis and Selection

We now have a set of complete solutions (that is, a series of capacity solutions, each associated with one or more acquisition methods). Prior to performing the financial analysis, the monthly cash streams the customer will have to bear over the useful life of the capacity solution are calculated for each capacity-acquisition pair. Cash streams are generated using information in the database on the various purchase and lease rate terms and conditions. Financial analysis is then performed for each of these plans. The analysis indicates how each of these plans will affect the customer's budget, tax books, and profit-and-loss books. The cash stream generation and financial-analysis methods used are fairly conventional accounting methods approved by the Internal Revenue Service (IRS) and the Financial Accounting Standards Board (FASB).

Using numbers from the financial analysis, selections, which are based on two criteria, can be made of the preferred solutions: financial considerations and qualitative considerations. *Financial considerations* dictate that the financing plan which is best from the pure finance point of view should be selected. *Qualitative considerations* go beyond the pure dollar consideration. For example, the customer's business interests might require the selection of a plan even if it is not the most attractive financially. Very often, financial and qualitative considerations can result in the selection of two different plans. The marketing representative thus needs convincing arguments for both, depending upon the person to whom the arguments are addressed as well as the person who ultimately makes the acquisition decision. A sample of these selection heuristics follows: The financial example is, "Propose the solution that has the best discounted after-tax cash flow if the customer contact point is a financial executive who utilizes discounting and is concerned with his cash flow." The qualitative example is, "Propose a lease solution if the key decision maker views his business as cash poor, with a low effective tax rate and a high borrowing rate."

These selections need to be presented in the form of convincing financial arguments, and a system that generates such arguments should be able to conduct a dialogue on the contents of the argument. The user might wish to see further explanation on a point or might require help on how to clarify some concerns. The system should be aware of the user's intentions in order to minimize actions required on the user's part and to maximize the utilization of the limited bandwidth (sales call) available for conducting this explanation process.

Now that we have discussed the domain in general, we describe the implementation of an initial prototype system. In this system, there is special emphasis on a mixed mode of interaction in which the user and the system cooperatively solve problems under the user's control, with guiding and focusing strategies provided by the system. This interactive operation mode, which allows the user complete control over the direction of problem solving but able to draw upon the system's expertise for guidance and planning, has successfully been used before in systems such as expert log analysis system (ELAS) (Apté and Weiss 1985) and VLSI expert editor (VEXED) (Mitchell, Steinberg, and Schulman 1985).

Initial Prototype System

The initial prototype system was written as a set of relatively independent OPS5 rule groups. Each rule group contains its own computational expertise and communicates with the other groups through a set of protocols. This grouping of the rules was originally intended to divide the system into small easily maintained subsystems, but as we point out, this grouping led to computational bottlenecks.

In the simplest case, communication between rule groups is accomplished by simply sharing internal memory. For other communication, a rule group generates the data expected by the next group and then creates that group's task, thereby transferring control. For example, access to the product-offering database (products, pricing, and lease rates) is achieved for each routine by creating an incomplete product working memory element:

```
(make db:product
   ↑product       3083
   ↑model         JX3)
```

The database rules detect that the prices are missing and fill in the rest of the data. The rules requiring pricing information for the 3083 JX3 do not fire until after the database query on that product is completed successfully

```
(db:product
      ↑product        3083
        ; Product number
      ↑model          JX3
        ; Model/FEA number
      ↑description    Processor Unit
        ; Description
      ↑purchase       1975000
        ; Purchase price
      ↑rental         128310
        ; Monthly rental
      ↑lease          102650
        ; Monthly lease
      ↑maintenance    3695
        ; Monthly maintenance
      . . . )
```
[2]

Alternatively, control can explicitly be passed when the necessary calculations have completed. The following is a rule that passes control from the capacity-analysis rule group to the explanation rule group:

```
(p cp:advance-goal-to-explain
   (dp:waiting ↑status <status>)
     ; If the computing
  {<goal>
     ; alternatives have
   (cp:goal
     ; all been
      ↑type calculate)}
     ; generated,
→
   (modify <goal>
       ; then proceed with
      ↑type explain)
       ;     the explanation by
   (make xs:start
       ;     setting explanation
      ↑goal start
       ;     goal. When done,
      ↑title Computing-Alternatives
       ;     the goal will be
      ↑return-to-caller explained))
       ;     modified to explained.
```

Figure 2 shows the major rule groups in the initial prototype system along with typical control flow paths.

The *customer information gathering* rules query the user or database to obtain information about the customer company, its computing needs, and its financial status. This

[2]Prices shown are merely for illustrative purposes and might not reflect actual pricing

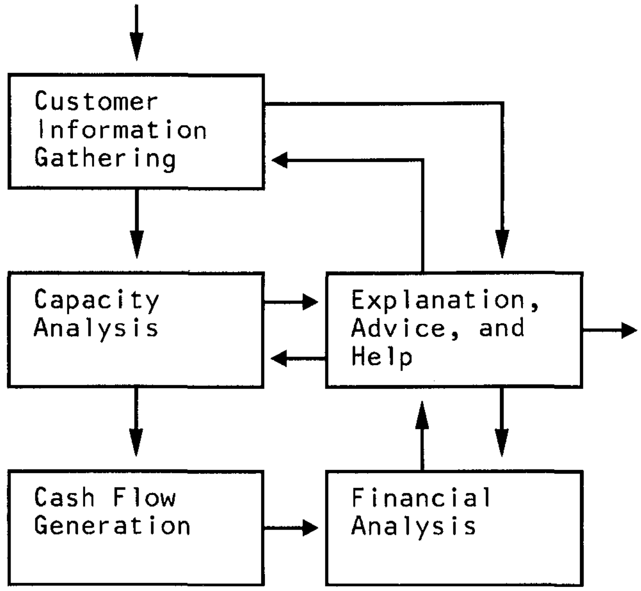

Figure 2. Rule Groupings with Typical Control Flow.

information is then given to the *capacity-analysis* rules that generate all reasonable computing alternatives for the given compound growth rate in computer capacity. This information is then given to the *explanation generation* rules that allow the user to verify, add to, or modify the computing alternatives to be considered in further analysis. Next, the *cash flow generation* rules translate each computing alternative into the monthly cost of that computing environment. This information is then fed to the *financial analysis* rules that provide a ranking of the alternatives. The *explanation generation* rules then use the information provided by the previous rule groups to provide explanations, justifications, and arguments for and against computing and financial alternatives. The next subsections describe each rule group in greater detail.

Customer Information Gathering

The first step in analyzing the current situation is the user input of relevant customer information. The system attempts to gather the following types of information: customer identification, analysis period, current install base, capacity planning, customer history, and current financial information.

Typically, the system is able to access either common databases (for example, the sales manual, corporate financial profiles, and current install base information) or databases generated from previous runs of the system to fill in most of the over 40 categories of customer information used in the initial prototype system. Therefore, a typical session requires only the identification of the customer, the period of time for this situation analysis, any recent changes to the customer information, and special constraints or restrictions for this analysis.

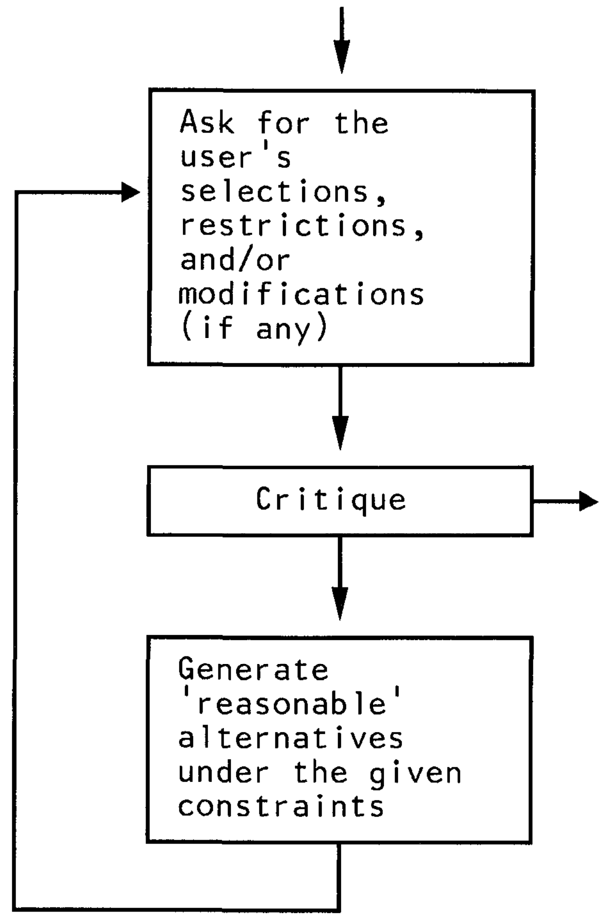

Figure 3. Capacity Analysis.

The rule group responsible for obtaining customer information is automatically invoked whenever the next stage of the analysis requires some input information that is not available within the system resources. Thus, the user can command the system to proceed with an analysis without first worrying whether the necessary information has been entered.

Capacity Analysis

The next step in a typical situation analysis is the generation of the possible computing options to fit the customer's capacity requirements and growth plan. Figure 3 shows the major steps in this rule group. The computing options rule group (together with the explanation rule group) provides the user with the capability of specifying as much or as little of the computing alternatives as desired. The system then critiques (by way of a one-line message) each user input alternative and generates all "reasonable" unspecified parts of the computing plan. The "reasonableness" criteria are encoded as rules and, hence, are quite flexible. Refer to the Capacity Planning subsection for three examples of such criteria.

For example, the user might provide the input that the 3083 JX3 is to be retained. The system critiques this information and indicates that the JX3 will lack the necessary processing power for its application (VM) on December 1985. However, this input is allowed, and the system proceeds with the analysis. Then the system generates all possible upgrades or replacements for the computers running the other two applications (MVS development and MVS production). The user can then further restrict the alternatives by eliminating some of those generated.

The combination of automatically generated alternatives within user-specified constraints provides a powerful tool for the novice and expert user alike. If the user already knows what options are to be considered, then they merely need to be input. Otherwise, the system is capable of generating what it considers are reasonable alternatives.

Cash Flow Generation

One of the most computationally intensive parts of the initial prototype system is the rule group that calculates the total cost of computing for each computing and financing alternative. This rule group computes month by month the principal, interest, annuity, insurance, maintenance, property taxes, and so forth, where applicable. In addition, these rules ensure that all payment streams start and end on the same dates for each alternative. This rule provides for a fair financial comparison of the alternatives.

For the TNW case, the capacity analysis was able to heuristically prune the list of alternatives to four, which meant that to consider two financing options over a four-year analysis of each of the computers in each alternative required over 1,100 rule firings ($2 \times 48 \times 3 \times 4$). We later recoded most of this analysis in Lisp.

As the domain coverage increases, this computational bottleneck becomes an even more serious problem. It is not uncommon for a large computing center to have tens of machines, with major components financed separately with different terms and conditions and with multiple viable upgrades and replacements. It has become obvious to us that this simple generate-and-test paradigm will shortly become intractable.

Financial Analysis

Figure 4 shows the major steps in the financial-analysis rule group. The strategic constraints for this analysis are determined from the information known about the company. Most of this information is gathered by the customer information gathering rule group. Next, the categories of payments from the cash flow generator rule group are combined to produce a picture of how each alternative affects the budget, the taxes, the profit-and-loss statement, and so on, of the company. Utilizing expert financial knowledge, the financial rule group strictly ranks each alternative financially (that is, which alternative is the least expensive) and qualitatively (that is, business judgment by the customer and the marketing representative). These rankings are purely heuristic and reflect our experts' opinions of successful marketing practices. The heuristic nature of the advice offered by this rule group demands the support of explanation capabilities in order to provide a useful tool.

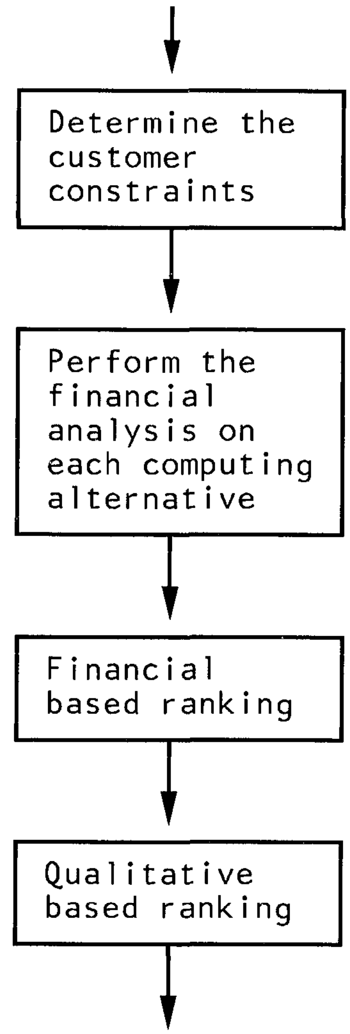

Figure 4 Financial Analysis.

Explanation, Advice, and Help

The explanation rule group assumes control of the session after each major step in the analysis. The default output gives a general explanation of the current state of the analysis, prompts for one of several options on how to proceed further with the analysis, and provides advice on the use of these options. The explanation rule group contains user-modeling and session-monitoring capabilities to anticipate the user's intentions, which enables the group to provide defaults for the user. In addition to the screen generation and control rules, the explanation rule group consists of rules for handling user intention, controlling advice, and controlling help.

The user intention handling rules use a set of explanation templates linked in a network by the possible queries on each template. The explanation templates are filled using information left in working memory by the rules that calculated the information being explained. For instance, the generation of computing alternatives includes the critiquing of alternatives that won't be considered in later calculations. This critique is then available in the explanation system so that the user can ask why a particular computing alternative wasn't considered.

The user can question any part of the explanation by merely pointing to its representation on the screen. The interpretation of that questioning is a context-dependent traversal of the explanation network to a new template. This template is then appropriately filled and displayed. In other words, the system is able to estimate the user's intention based on the session history and the part of the screen pointed to. The system can do this because we were able to anticipate potential queries from the user for further information. This user modeling works well for our domain because we were able to effectively determine *a priori* the typical questions.

The rules that control advice use the explanation network to guide the user to the next recommended action: The cursor is placed on the screen at the point of the explanation that (according to our studies) would typically be probed next. The rules that control help also use the explanation network to provide a small pop-up window at the bottom of the screen to indicate what operations might be desirable in addition to what is recommended by the current cursor position.

Each screen displayed by the explanation rule group is represented internally as one OPS5 working memory element. The attribute-value pairs contain information about the various slots in the explanation template. When a screen is being prepared for display, all appropriate slots in the working memory element for that screen are filled by rules which collect information left in working memory by the problem-solver rules. The screen is then displayed, and the user is given the opportunity to place the cursor over any of the filled-in items in the template. When the ENTER key is pressed, the cursor position is read, and the information is placed in the working memory element for that screen. Rules from the explanation rule group then interpret the cursor position as an indication that the user is requesting more information on the item which was under the cursor.

Associated with each selectable item on an explanation template is an ordered collection of pointers to other explanation templates. In other words, each selectable item on the screen points to all of the available screens which could possibly be used to further explain that item. The ordering of the other explanation templates is context dependent. This ordering enables the system to provide one explanation (the most commonly asked for) the first time an item is selected and at a later time provide another explanation. Thus, each time the user places the cursor on an item a different explanation is possible.

We provide two examples of the explanation capability. Figure 5 shows a "Recommendation" screen that indicates a

```
Recommendation
We recommend that on May 1986, TNW take the following action:
upgrade the 3081 K32 to a 3084 Q64 for MVS-PRODUCTION
and finance it using PURCHASE

The Cumulative Cash Flow After Tax over 48 months resulting from
the above recommendation is $10,000,000, which is the minimum
among possible alternatives   It is also the best alternative with
respect to Cumulative P&L Impact

The Cumulative Cash Flow After Tax is $123,400
less than that for the next best alternative, which is to
upgrade the 3081 K32 to a 3084 Q64 for MVS-PRODUCTION on May 1986
and finance it using IBM-LEASE

→ Go to Financial Analysis
→ Go to next screen

This recommendation is based on purely financial consideration
Advice
Though the above recommendation seems appropriate for TNW
from a financial point of view, you may wish to take into account
a qualitative consideration.  Press Enter to see the next screen
Press PF3 to quit, PF2 for help
```

Initially the cursor is located on → Go to next screen

Figure 5. An Example of the Initial Screen of Explanation.

```
Recommendation
  Date for Action to be taken
  The reason why May 1986 was chosen is

  On May 1986, the 3081 K32 running MVS-PRODUCTION
  will lack the needed performance.

  → Go to Graph of Performance needed
  → Go to Graph of Memory needed

  If you want to change the date,
  → Go to capacity requirements screen, or
  → Go to computing alternatives screen
  Advice
  ...
```

New screen is overlaid on the previous one but displaced

Figure 6. Telescopic Screens.

recommendation derived as the result of a capacity analysis and a financial analysis.

At this point, the user can press the ENTER key to go to the next screen. In this case, a screen entitled "Recommendation with qualitative consideration" is displayed. Alternatively, the user can position the cursor at any point in the text and press ENTER to provide a detailed explanation based on the particular text. Several areas in the text can be colored to emphasize their importance. For example, if the cursor is placed on the date "May 1986," before pressing ENTER, a screen entitled "Date for Action to be taken" is shown, overlaying the previous screen in the telescoping fashion shown in figure 6.

This explanation facility was quite effective for our initial prototype system because the user was not required to know very much about the domain (every screen includes a pop-up help window), a standard analysis path is always available to the user (the next selection is under the cursor by default), and typing is kept to a minimum. However, the user is always in control of the session because the system is merely providing defaults. It remains to be seen if such a simple facility can be as effective when the domain coverage and complexity increase.

Current Implementation

The initial prototype system was written in OPS5, which in turn was written in Lisp/VM. The system mainly consists of OPS5 rules with supporting Lisp functions for efficient database manipulation, graphics handling, and some numerical calculations. All of the system's control and inference processes are encoded in rules. This makes the system quite flexible. The following is a breakdown of the rule groups:

Number of Rules	Description
7	Top Level Interface and Control
50	Graphic Display Handler and Window Manager
40	System Documentation
16	Database Interface (e g sales manual)
120	Financial Analysis
147	Customer Information Gathering
40	Capacity Analysis
18	Cash Flow Generation
337	Explanation, Advice, and Help
775	Total number of rules

On the average, there are 4 condition elements (if clauses) in each rule, about 3.5 attributes in each condition element, and 4 action elements (then clauses) in each rule. The TNW scenario analysis requires approximately 3,000 rule firings and uses in excess of 11,000 working memory elements.

The financial marketing initial prototype system rule set consists of slightly more complex rules than our group's previous effort, YES/MVS (Griesmer et al. 1984; Ennis et al. 1986). Unlike YES/MVS, the FAME initial prototype system rules often set up long complex chains of goals requiring substantial numeric calculations. YES/MVS consisted of rules to diagnose problems in a computer center and to perform remedial actions. Thus, many of the rules classified problems into a known set for which there was a fixed set of remedies. The financial marketing initial prototype system, however, consists of rules to generate many possible alternatives (computing and financial) and to provide reasonable explanations for these alternatives. Also, the interactive explanation capability accounts for a substantial portion of the rule set.

Ongoing Research

Using the experience gained in our initial prototyping effort, we are currently developing new representations and algorithms for a complete restructuring of our system. We are attempting to generalize from our early efforts to build a financial marketing knowledge-based consultant for the financial marketing domain in order to provide a set of generally useful mechanisms for similar domains.

Several of the rule groups involved in the generate-and-test paradigm pose a serious computational bottleneck. The use of clever heuristics to prune the search space will probably have limited usefulness with the present representation scheme. We are currently attempting to utilize structured inheritance networks (Mays and Balzac 1986) to represent the shared structures of the domain, which are so often found during the development of the initial prototype system. The inheritance mechanisms will allow us to share computation by doing computation on abstract aggregate objects rather than on each instance.

The financial calculations performed in the initial prototype system were spread throughout the rules. This style produces a substantial rule maintenance problem. The dynamic nature of the financial world will constantly force changes in the way financial calculations and their results are made and reported. We have begun work on the fundamental modeling of financial equations (Apté and Hong 1986) so that the system will have one consistent representation from which the various uses can be derived.

The initial prototype system uses a relatively fixed computational sequence. Each rule group works relatively independently on its task. However, we have found that financial marketing experts opportunistically mix the solution of aspects of the capacity-planning and financial-acquisition problems. This mixing is usually driven by customer constraints. We are currently evaluating blackboard models to allow such opportunistic problem-solving planning (Hayes-Roth 1985).

Summary

The initial prototype system solves a small subset of the problems in the financial marketing domain but not in as much detail and depth as the system currently being developed to replace it. At present, the initial prototype's domain coverage is somewhat spotty and severely limited. Currently, it only addresses problems of a selected subset of large mainframe processors, their upgrades, and replacements and can only compute the financial impact of straight lease and outright purchase acquisition methods. Even though severely limited in its scope, the initial prototype system generated an enthusiastic response from our marketing organizations. The domain spectrum will be broadened as we introduce more product lines, such as disk drives, terminals, and other processor and peripheral families, as well as other financing options. Modeling of situations will also become comprehensive and detailed as more assessment data are introduced.

The financial marketing application has revealed itself to be a rich vehicle for working on a number of interesting open problems in AI and expert systems, and the completeness and usefulness of the final system will be impacted significantly by the research in strategy modeling, representation and control in planning, knowledge acquisition, and explanation generation and the development of models of man-machine interaction and of intelligent interfaces to existing analytical tools and databases. The next version of the FAME system, based on the principles and methodologies developed after the initial prototyping effort, will be tested on real cases. Such a productivity tool could have considerable impact on the ability of marketing representatives to market their products effectively in a rapidly changing business world.

Acknowledgments

We are indebted to numerous people from the IBM marketing division for providing expertise and guidance on this project.

References

Apté, C V , and Hong, S J 1986 Using Qualitative Reasoning to Understand Financial Arithmetic In Proceedings of the 1986 National Conference on Artificial Intelligence, 942-948 Menlo Park, Calif : American Association for Artificial Intelligence

Apté, C. V , and Weiss, S M 1985 An Approach to Expert Control of Interactive Software Systems *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(5): 586-591

Ennis, R L ; Griesmer, J H ; Hong, S J ; Karnaugh, M ; Kastner, J K ; Klein, D A ; Milliken, K R ; Schor, M. I ; and VanWoerkom, H M 1986 A Continuous Real-Time Expert System for Computer Operations. *IBM Journal of Research and Development* 30(1): 14-28

Griesmer, J H ; Hong, S J.; Karnaugh, M ; Kastner, J K ; Schor, M I ; Ennis, R L ; Klein, D A ; Milliken, K R ; and VanWoerkom, H M 1984 YES/MVS: A Continuous Real-Time Expert System In Proceedings of the 1984 National Conference on Artificial Intelligence, 130-136. Menlo Park, Calif : American Association for Artificial Intelligence

Hayes-Roth, B 1985 A Blackboard Architecture for Control *Artificial Intelligence* 26:251-321

Lindsay, R ; Buchanan, B G ; Feigenbaum, E A ; and Lederberg, J 1980. *Applications of Artificial Intelligence for Chemical Inference The DENDRAL Project* New York: McGraw-Hill.

Mays, E , and Balzac, S R 1986 Interactive Reclassification in Structured Inheritance Networks Forthcoming

Mitchell, T. M ; Steinberg, L I ; and Shulman, J S 1985 A Knowledge-Based Approach to Design. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(5): 502-510

McCarty, L T , and Sridharan, N S 1981 The Representation of an Evolving System of Legal Concepts II: Prototypes and Deformations, Technical Report, LRP-TR-11, Laboratory of Computer Science Research, Rutgers Univ

Stefik, M 1981 Planning with Constraints (MOLGEN: Part 1) and Planning and Meta-Planning (MOLGEN: Part 2) *Artificial Intelligence* 16(2): 111-139, 141-169

AAAI-87 Workshop Request for Proposals

The AAAI-87 Program Committee invites members to submit proposals for the Workshop Program—expected to be an important feature of this year's conference.

Gathering in an informal setting, workshop participants will have the opportunity to meet and discuss issues with a selected focus. This format will provide for active exchange among researchers and practitioners on topics of mutual interest. Members from all segments of the AI community are encouraged to submit proposals for review by the committee.

To encourage interaction and a broad exchange of ideas, the workshops will be kept small Attendance will be limited to active participants only. Workshop sessions will consist of individual presentations, and ample time will be allotted for general discussion.

Please submit your workshop proposals to:

Joseph Katz
MITRE MS-D070
Burlington Road
Bedford, Massachusetts 01730
Katz@mitre.arpa