ANALYST
An Advisor for Financial Analysis of Automobile Dealerships

Michael A. Hutson

General Motors Acceptance Corporation, Forward Systems, 3044 W. Grand Blvd., Detroit, Michigan 48202

Abstract

ANALYST is deployed in 230 GMAC domestic branch offices. There, each year, 400 credit analysts perform nearly 20,000 financial analyses of automobile dealerships. Using ANALYST they will save an estimated 55,000 man-hours. The focus of this knowledge-based system, from the very beginning, was its end user -- a credit analyst. Driven by singular end user requirements, developers extended the native AI tool explanation facilities, built an easy to use interface and completed the integration of engineering workstations in GMAC's IBM computing environment.

The Application Domain

One of the services GMAC offers to approximately 12,000 domestic General Motors dealerships (and affiliates) is inventory financing. This service is also known as wholesale or floor plan financing. In exchange for funds, dealers must adhere to a set of rules -- the most important of which is to pay GMAC promptly as vehicles are sold. This adherence is analyzed at least annually.

The process of analyzing a dealership is, in essence, financial ***risk*** analysis -- not because dealerships are risky businesses, but because borrowing and lending money always contains an element of risk. In a typical analysis a GMAC credit analyst evaluates the risks by examining the dealership's past performance, local economy and operating ability. The credit analyst then predicts the dealership's likely performance until the next scheduled review. Next, the credit analyst recommends credit lines and suggests ways to reduce risk. Finally, the credit analyst's recommendations are adopted or changed or both by management.

The Problems

To be successful, ANALYST must address a number of domain-related problems. First, financial analysis procedures are time consuming. A single analysis ranges from several hours to a few days depending on the complexity and cooperation of a dealership. On average, however, each review requires between six and seven hours to complete. Contributing to the length of time credit analysts must spend is the nature of the data: financial statement data is voluminous, prone to errors and, in a small percentage of the cases, deliberately misrepresented. This data issue is significant. Its effects are pervasive; from knowledge representation to user interface.

Second, risk analysis expertise requires years of experience as well as a thorough understanding of accounting principles and financial analysis theory. Most entry level employes do not have educations in accounting or finance. Therefore, new analysts tend to concentrate on the more mundane tasks of the process, for example, calculations of ratios. In addition, although credit analysts quickly become proficient in following established procedures and identifying out-of-guideline situations, they frequently overlook danger signals with respect to the financial condition of a dealership. This problem is exacerbated by a declining experience level in the credit analyst population as a whole. As management retirements increased and business grew (notably in the last five years), numerous promotions from the ranks occurred. The ramifications of this declining experience level problem are that some novice credit analysts tend to be too lenient, because they perceive they will lose wholesale accounts to less restrictive GMAC competitors. Conversely, some novice analysts, by being overly restrictive, drive high-performing dealerships to competitors.

Last to be discussed here, the third problem is a common one for many businesses: computer-illiterate users. While nearly all GMAC credit analysts have experience with 327x ("dumb") terminals, few have used a personal computer and none, prior to this project, had touched an engineering workstation.

ANALYST in Depth

Logical System Architecture

ANALYST uses well over a thousand data elements during a review, and allows the credit analyst to display several thousand more in forms, tables and graphs. Most of this data originates from geographically distributed GM and GMAC databases, some of which are not available online.

The central object of the analysis process is the dealer-prepared financial statement. For GM dealerships, this is a 4-page form containing a balance sheet, income and expense summary, and detailed information about departmental sales, expenses and inventories. Most GM dealerships subscribe to a GM service which prepares monthly financial statements from trial balances supplied by the dealerships. ANALYST accesses this information directly when so authorized by dealerships. Information about GMAC's experience (rating, credit lines, special programs, monthly retail and wholesale history) with the dealerships is also used during a review. This information is maintained in online databases.

Off-line feeds to ANALYST from these sources occur nightly or monthly, depending on the source. This data is massaged, organized by dealership, and stored in a central database. Data is kept from the two most recent financial statements so that credit analysts have the option of reviewing earlier information. In addition, information from the previously reviewed and previous year-end financial statements is retained for use during subsequent reviews. Financial data not available from online sources is entered clerically. Actual review processing is done by the knowledge-based system which runs on the Teknowledge, Inc. S.1 tool (figure 1).

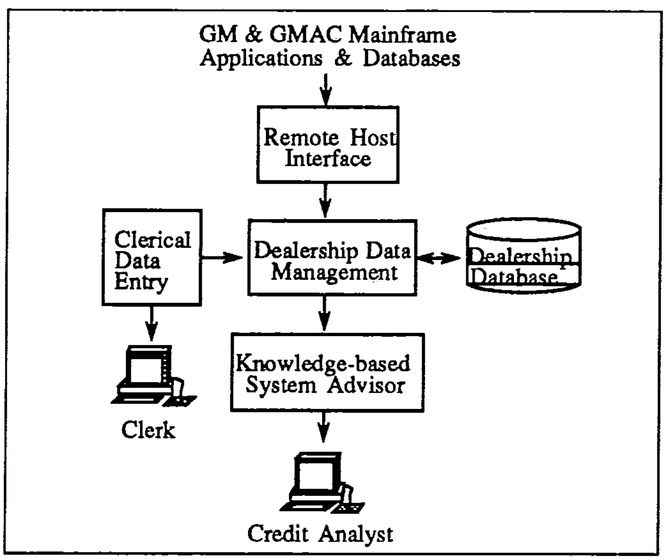

Figure 1. Major processing components of the logical system

Physical Architecture

ANALYST was deployed as distributed components. Financial statement information is extracted nightly from an IMS database. GMAC experience data is extracted monthly from DASD resources. Receipt, processing, storage (in a VSAM file) and preparation for distribution of data are performed by batch jobs that run on an IBM mainframe operating under MVS.

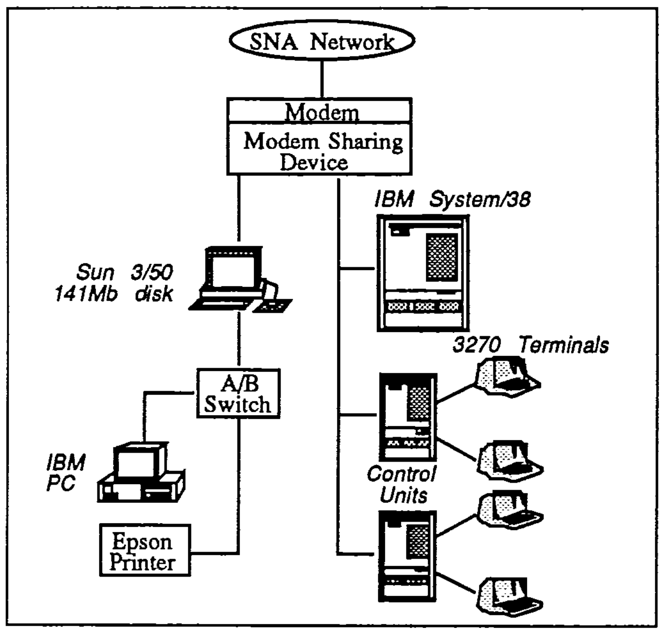

Figure 2. ANALYST branch hardware configuration

The knowledge bases and clerical data entry components reside in a single Sun Microsystems workstation located at each GMAC branch office. All communications between the workstation and mainframes are performed over an SNA network. To allay concerns about end user response time and network performance, data is transferred between the host and workstations at night. Completing the process, downloaded data is stored in a local database until the review is completed.

The workstation configuration at the branches is shown in figure 2. The basic branch workstation is a

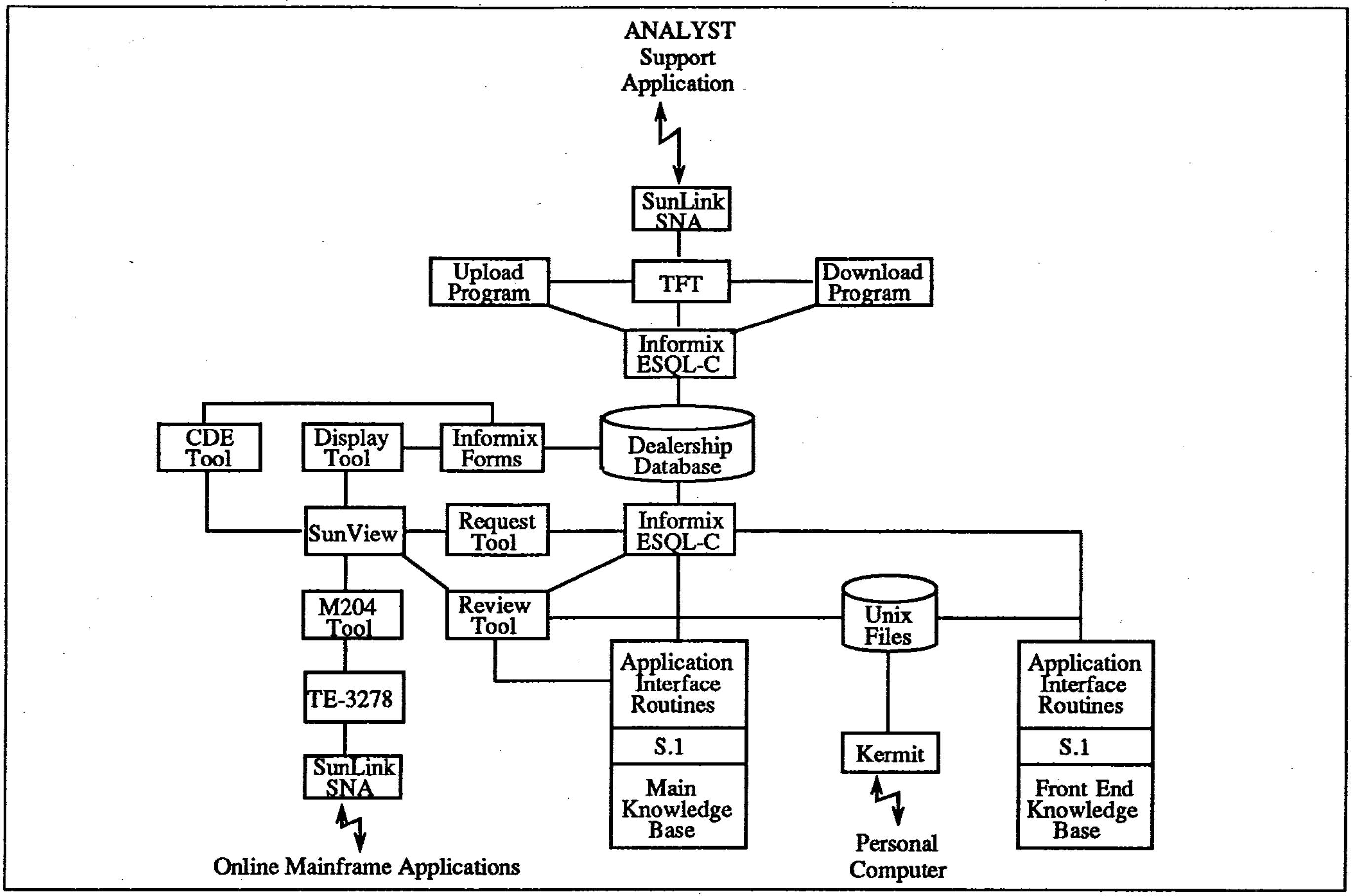

Figure 3. Workstation Platform Software

Sun 3/50 with a standard 19" monochrome display and a 141 Mbyte disk. One workstation serial port is connected to a modem sharing device, which along with one or more branch control units, accesses the SNA network. The other serial port is connected to an A/B line switch which permits the workstation to alternately access a personal computer and printer.

The Workstation Platform Software

To help understand the branch workstation software platform, a high level look is shown in figure 3. Dealership data is stored on an Informix database, with software access provided through Informix ESQL-C facilities. The Informix forms facility presents full-screen displays of the dealership's four-page financial statement. The same form is used to clerically enter dealership financial statement information.

SunLink SNA facilities transfer dealership data (and requests for data) between the host and workstation and, interactively, access online host applications. The unattended file transfer operations use TFT, the Transparent File Transfer facility developed by Electronic Data Systems Corporation (EDS). TFT uses the IBM FTP capability of Sun's TE3278 facility. Credit analysts can access mainframe applications through the M204 tool, which uses TE3278 to provide the 327x terminal emulation.

The User Interface

To make ANALYST easy to use, workstation functions are organized as an extendable set of application "tools." All the functions run under native SunTools, with separate icons and windows (figure 4) which are displayed after a credit analyst logs in.

Furthermore, the user interface is the "glue" that joins all of ANALYST's major workstation components: knowledge bases, S.1, Informix database and 327x terminal emulation. The user interface, conceptually, is organized into layers. The top layer is a program (actually, just a UNIX shell script) which starts

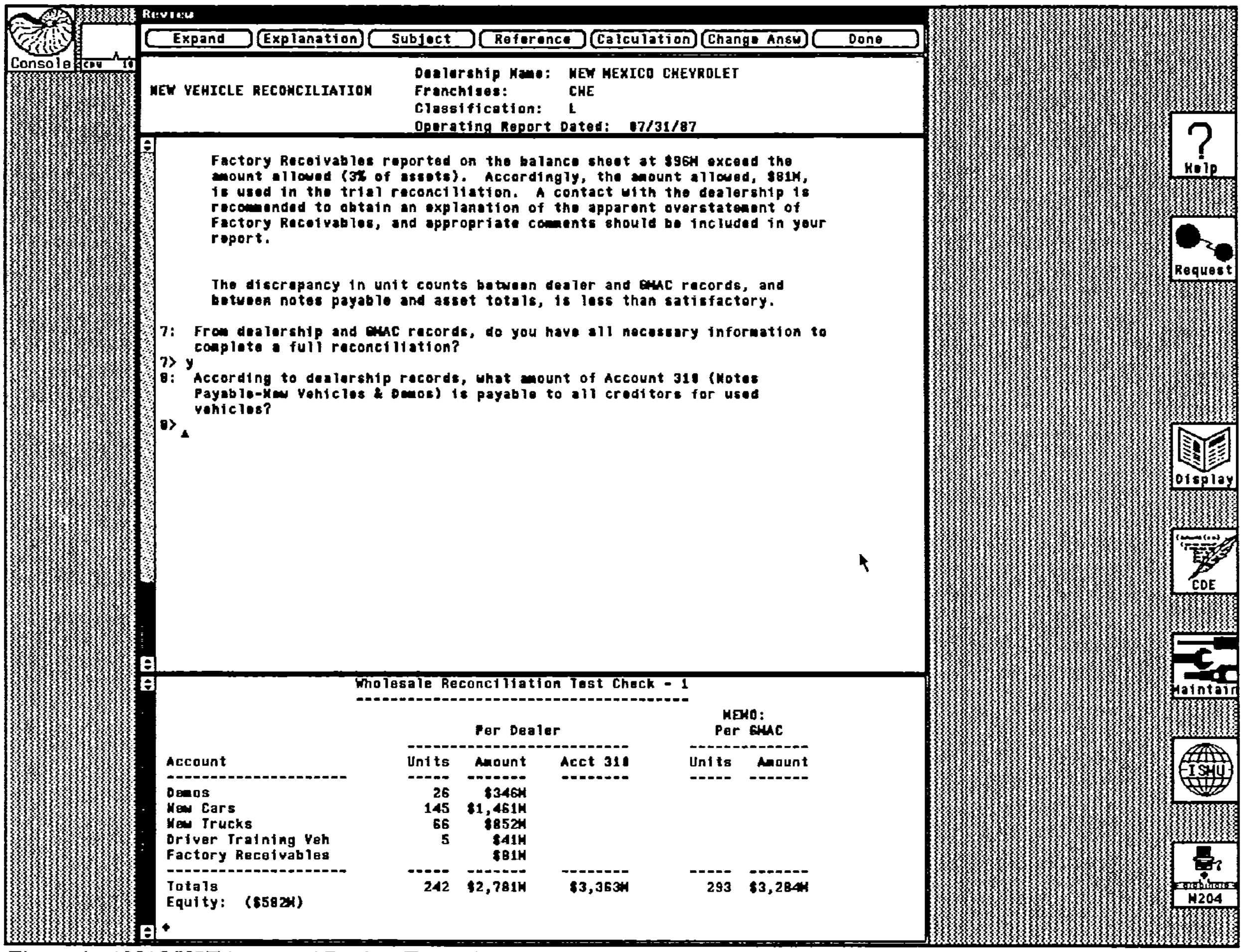

Figure 4. ANALYST icons and Review Tool window

up -- in the background -- each of the tools. The second layer, is comprised of the tools themselves, each consisting of a base frame and screen(s). Each screen is composed of a set of building blocks which comprise the bottom layer.

Knowledge Representation

Expert dealership risk analysis combines data abstraction and evaluation, association of problems with corrective actions, and refinement of recommendations (figure 5). In the initial step, each significant piece of financial statement data (or calculation involving several financial statement elements) is abstracted into more significant data features. Implemented in the form of rules and constraints, this step allows the expert to reason with symbolic rather than numeric data.

The initial step also involves an evaluation of the numeric data; the symbolic abstractions generally indicate when the financial statement data (or calculation result)

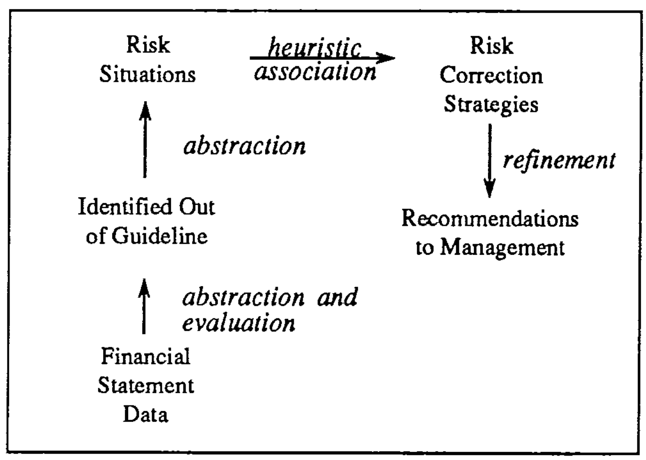

Figure 5. Knowledge Representation Steps

is out of guidelines. For example, the values concerning used car inventory level and rate of sales become statements about the dealership's inventory status. Thus, a 73-day supply of used cars may be abstracted as an "overstock of used cars" (figure 6).

Once the data is in symbolic form, the credit analyst interprets the data characteristics and further abstracts out-of-guideline situations into risk situations. Reasoning, here, is accomplished with more complex rules (because more judgement is involved) than those in the initial abstraction step. Continuing the example, further analysis may conclude that the dealership (and GMAC) is at risk because of the excessive investment in used car inventory.

In the next step, risk situations that have been identified are associated with potential corrective strategies. For example, if a dealership has an overstocked used car inventory, one recommendation is to liquidate the inventory.

In the final step, the recommendation is refined using additional analysis and, possibly, requiring the analyst to gather more information. Completing the example, selling to wholesalers just the used cars on hand for over thirty days may remedy the risk situation.

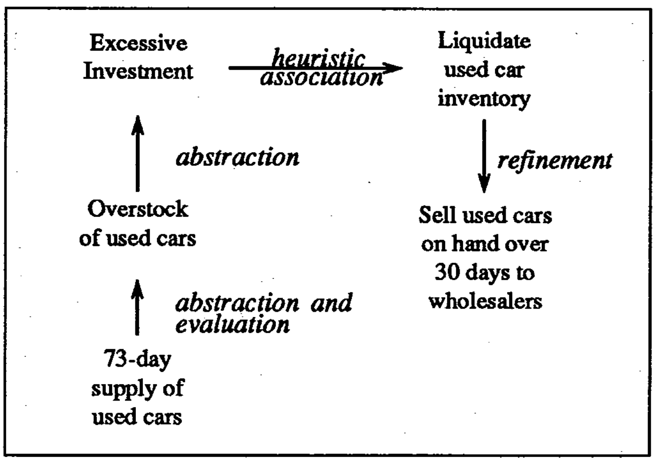

Figure 6. Knowledge Representation Example

The Knowledge Bases

In the original design of ANALYST, there was only one knowledge base. As the system matured, it became evident that credit analysis knowledge could be used in three different ways: (1) to verify that the clerically entered data met a minimum set of requirements, (2) to perform a preliminary analysis on the data received from the host before a consultation commenced, and (3) to conduct an interactive consultation with the branch credit analyst. Accordingly, three knowledge bases were developed. In this article, only two of the knowledge bases are discussed: the frontend (non-interactive) and main (interactive).

As mentioned earlier, data reliability is a concern of credit analysts. Since data is the starting point in knowledge representation (figure 5), the frontend performs the important function of *advising* the credit analyst about missing, inconsistent and questionable data.

The frontend knowledge base was designed to remove data validation and routine calculations from the main knowledge base. The advantages of this design are a smaller, more maintainable main knowledge base and faster running consultation. Again, rules and constraints were used to model the expert's reasoning. Among the activities of the frontend are:

- perform calculations and check the sensibility of the results
- perform screening tests to check for the validity of balance sheet items
- check to see if the two tests above yield significant problems
- compare the current and previous financial statements for significant changes
- summarize and print the above findings
- set an intensity level that determines the depth of questioning during the consultation.

As an example of a frontend activity, setting an intensity level is achieved by determining the attribute, *high.anxiety?[r]*, which indicates that the dealership may be in trouble. This boolean attribute is set only when one of several rules is fired. The control block *det.high.anxiety?* controls the processing including storage of the results in the local database. The control block, attribute and a typical rule are shown below.

```
DEFINE CONTROL.BLOCK det.high.anxiety?
/* fe.general.ctl */
::ARGUMENTS  a:attribute, r:review
::INVOCATION  determination
::BODY    begin
          vars dummy.b:boolean
          seek a[r] by rules;
          if high.anxiety?[r] known then
             dummy.b:=db.put.b(high_anxiety,
             high.anxiety?[r])
          else dummy.b:=db.put.null (high_anxiety);
          end
END.DEFINE
```

```
DEFINE ATTRIBUTE  high.anxiety?
/* fe.general.ctl */
::DEFINED.ON       r:review
::TYPE             boolean
::MULTIVALUED false
::LEGAL.MEANS {try.rules}
:: DETERMINATION.BLOCK  det.high.anxiety?
END.DEFINE
```

```
DEFINE RULE   rul.intensity.net.loss
/* fe.general.rul */
::APPLIED.TO        r:review
::PREMISE           (db.get.i(net_profit_aft_tax) < 0
::CONCLUSION      high.anxiety?[r]
::JUSTIFICATION "if there is a net loss after bonuses
        and taxes, this review should be conducted at the
        maximum level of intensity."
END.DEFINE
```

The main knowledge base runs the consultation and produces the review reports. It contains over 3,500 objects organized into layers. The organization of this knowledge base was partly motivated by the observation that a large amount of the area-specific analysis, (for example, cash, new vehicle inventory) is nearly independent of other areas. Further, the same kinds of processing were done for each area. Although S.1 does not provide an OOPS capability, a general object-oriented approach was chosen for the high-level control design to achieve simplification in development, maintenance, future extension, and -- to appease end users -- enhanced explanation facilities.

The highest level consultation control blocks are primarily concerned with the sequence of processing as perceived by the user. The sequence of events in the top most control block is:

1) Create the class instance for the review data
2) Get the name of the dealership to review (passed in from the user interface
3) Fetch the associated dealership data row, and determine whether or not the user wants to review that dealer (during which the results the frontend are displayed)
4) If the user accepts the validity of the data and elects to proceed, the *initiate.review* control block is initiated. It controls the following processing:
    - determine from the user which type of review to perform
    - perform specialized processing as dictated by the review type
    - determine whether the review should be conducted at a high or low level of detail (intensity)
    - save the reason for review in the database for later uploading to the host
    - invoke the review type-specific control block that directs the sequence of consultation "phases" to be performed
5) Update the database

Subsequently, the conduct of the consultation is determined by one of the eight review types. In turn, they define which area-specific analyses are to be performed. For example, the beginning section of the *regularly.scheduled.review* control block looks like this:

```
DEFINE CONTROL.BLOCK regularly.scheduled.review
/* top.level.ctl */
/* This control block is invoked from initiate.review in the
    case of a regularly scheduled formal review. */
::ARGUMENTS     r:review
::TRANSLATION  "control regularly scheduled review
                        sequencing:
::INVOCATION   internal
::BODY      begin
            invoke analyze.character.phase (r);
            invoke reconcile.ws.inventory.phase (r);
            invoke analyze.profit.and.loss.phase (r);
                        (and so on...)
```

Knowledge base layers comprise the rest of the framework, beginning with "phases" (figure 7). *Phases* reflect the general nature of the processing currently being done (for example, profit and loss analysis). The current phase is always displayed on the screen to provide a frame of reference for the credit analyst. Phase control blocks conditionally invoke *subphases* which are relevant to the type of review (for example, regularly scheduled, dealer loan request) to be performed. Subphase control blocks, in turn, invoke the area-specific analysis control blocks, *topics*. Three forms of topic-level control blocks execute approximately ten different functions including perform analysis ('analyze topic'), determine recommendations ('resolve topic') and generate reports ('print topic'). Invoked by topics, *task*-level control blocks perform a specific set of activities, some of which require user interaction. The scope of

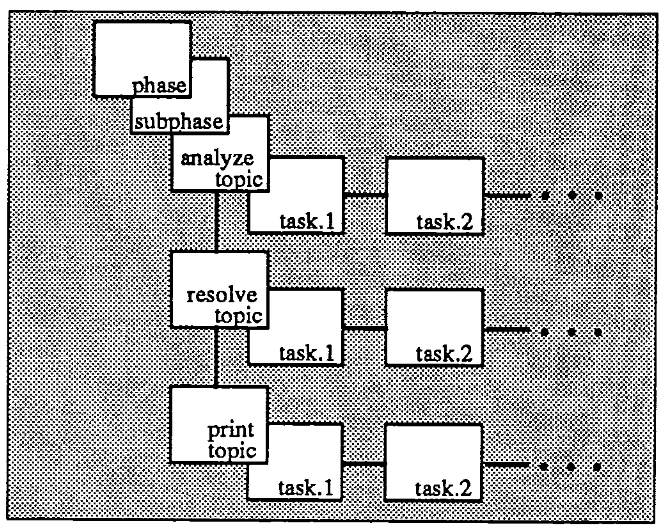

Figure 7. High-level main knowledge base structure

each task is limited so that explanations available to the end user make sense. Because explanations are a critical and unique aspect of ANALYST's delivered capabilities, they are discussed in greater detail below.

Explanation Facilities

The challenge in providing useful explanations was the credit analysts themselves; some would be very experienced and others complete novices. To meet the needs of both (and those in-between), multiple types of layered explanations were created in two broad categories: context-sensitive and context-independent.

All explanations are canned text located in a subdirectory on the workstation. These text files were created and are maintained solely by GMAC, which considered the native S.1 explanation facilities insufficient for end users. In the examples below, compare S.1 (top) and ANALYST text when executing a "why" command:

"I asked about the date of the last wholesale inventory in order to apply an.wholesale.inventory.since.review?.1. An.wholesale.inventory.since.review?.1 is used to determine whether there has been a wholesale inventory completed since the last formal review."

"The most fundamental asset on a dealership's lot is the inventory the business sells. Some dealers own this inventory outright, but most of them finance their new vehicles, and GMAC provides that financing.
Unfortunately, the process of reconciling the new vehicle inventory asset against the corresponding liability is complicated by the many different accounts the dealer can use in the day to day flow of business. But I can help you sort through this complicated process, provided you obtain all the necessary information for the wholesale reconciliation."

Explanations are available only when a task-level control block contains interactive code. In other words, the consultation must be at a question in order for a credit analyst to access any explanation type. From the user's standpoint, all explanations are obtained simply by clicking a button or using a pull-down menu. Internally, the user interface pipes a command string to an S.1 external function which retrieves the text and places it in the correct window. The mechanisms for retrieving the canned text differ by the type of explanation sought.

Context-sensitive explanations are tied either to an attribute (table 1) or to the structure of the knowledge base (table 2).

Type	Functionality (Answers the question:)
Help	What do I do next? What are the implications of my answer
Theory	What accounting theory (or principles are at play?
Syntax	What is the set of legal answers to the current question?

Table 1. Context-sensitive explanations tied to attributes.

Text file location and "layering" are achieved by simply adding a unique file name extension ('.hlp1', '.hlp2', '.hlp3') to an attribute name. As the user asks for additional help, the S.1 external function routine increments a counter and retrieves the appropriate file from the explanations directory. Three levels of *Help* are available for every question, each level containing more detailed explanations. Only one layer is implemented for *Theory* and *Syntax*.

Type	Functionality (Answers the question:)
Why	What am I doing here? What is going on at this point?
Situation	How did I get here?

Table 2. Context-sensitive explanations tied to kb structure.

The mechanism for locating text differs for the next two explanation types. Among the arguments passed to a task-level control block are a Why and a Situation *key*. These keys provide the indices for looking up the canned text. A stack is kept for each of these two explanation types. So, as lower level tasks are invoked, keys are stacked. When a credit analyst repeats the need for one of these explanation types, a key is removed from the stack. Therefore, each successive request returns an explanation more general than the previous one -- the knowledge base structure is climbed from bottom to top.

Context-independent explanations are shown in table 3. For these explanations, the text retrieval method differs slightly because knowledge base attributes are not involved. Specifically, the credit analyst determines (by selecting an item from a pull-down menu or by highlighting text in the consultation) the "name" used as an argument to the command string which is passed to the S.1 external function.

Type	Functionality
Subject	Explains key business concepts in the form of definitions of items such as materiality, intangible assets
Reference	Provides online access to GMAC operations and credit analysis manuals
Calculations	Displays the general formula, instantiated formula for the dealer under review and the result of the calculation

Table 3. Context-independent explanations

Project Management

Expected Payoff

For a dealer review application to be considered successful, GMAC expected to accrue a number of tangible and intangible benefits. The minimum payoff from tangible benefits was expected to be $2 million per year. Depending on the ability of ANALYST to reduce dealer losses due to default, the utilization of the system and the state of the economy, the system should increase the payoff.

Tangible Benefits:

- 50% reduction in time to perform reviews
- reduced effort for internal compliance audits
- elimination of planned costly training course
- reduction in losses due to dealer default

Intangible Benefits:

- standardization of the dealer review process
- thorough, consistent analysis
- enhanced performance of credit analysts
- additional time for the more difficult reviews
- value-added service to dealers
- method for training credit analysts
- a precursor for the ability to:
    - analyze any automobile dealership
    - analyze other types of businesses
    - analyze mega-dealerships
    - construct models of business types
    - market certain analysis modules
- ability to incorporate new analysis techniques

Project Life Cycle

Late in 1985, GMAC focused on an AI solution to automating the dealer review process. The study expanded when GMAC approached Teknowledge, Inc. of Palo Alto, California. Following conceptual and engineering validations, prototype development began.

At the completion of the prototype in December, 1986, GMAC conducted numerous demonstrations of ANALYST for its executives, managers and credit analysts. Because of the positive feedback from the demonstrations and from written evaluations of the system, knowledge base development resumed in January 1987.

System engineering work was undertaken by GMAC, Teknowledge, EDS, and American Management Systems (AMS) of Alexandria, Virginia in March, 1987. The following January, the system was installed in a single pilot branch. Eight branches were added to the pilot program in June, 1988. National rollout commenced during July at the rate of 16 branches per week. By December, all 230 systems were installed and operational.

Results

At the time of this writing, credit analysts had already performed 12,000 dealer reviews using ANALYST. Feedback from them over the months has been particularly rewarding. Although generally inexperienced with windows and mice, credit analysts quickly overcame their initial apprehension. They think the system *is* easy to learn and use. Beyond their familiarization with the personality of ANALYST, the normal learning curve for a good understanding of system concepts seems to be about 3 weeks. Then, users are comfortable enough with the system to trust it -- and ask questions about its logic.

Objective feedback was obtained too. In a recent branch survey users verified that many of the intangible benefits were being met -- some exceeded. Although the full tangible impact of the system cannot be realized for several months, most users indicated that the system was already providing measurable time savings.

Acknowledgments

I am particularly grateful to a unique group of hard working people -- "ANALYST Survivors" -- without whom this project would still be in development and have been less fun. You know who you are. And, this project is a credit to GMAC executives, who were willing to take a fair amount of risk of their own.