FRAMENTEC

M.1

ARTIFICIAL INTELLIGENCE ON THE PC

M.1

Overview

M.1 is a powerful, yet easy to use knowledge engineering tool for personal computers. M.1 is specifically used to design, develop and run stand-alone expert systems.

These systems are created by building a knowledge base containing facts and rules about a chosen application or problem domain in a special English-like language. After a knowledge base is constructed, M.1 can engage a user in a question-and-answer dialog, about a specific problem. M.1 uses the knowledge base to reason about the problem and make a recommendation or conclusion. During or after a consultation, the user can request explanations of the system's reasoning process, or ask the system to justify its conclusions.

M.1 is designed for rapid prototyping of potential full-scale operational systems.

M.1 is also appropriate for developing operational systems, with over 200 knowledge base entries to be loaded at any one time.

M.1 is the first knowledge engineering tool with significant functionality engineered for the microcomputer environment.

M.1 contains advanced features such as a multi-window display, interactive knowledge base debugging, automatic question generation, legal response checking, explanation facility, certainty factors, and the use of variables in the knowledge representation language. To date, systems containing such features have typically required specialized or dedicated hardware, or knowledge of a symbolic programming language.

M.1 is easy to learn and use, and can be mastered by people with programming experience. It does not require previous knowledge engineering background. By providing extensive functionality and exceptional ease of use in an inexpensive package, M.1 represents a breakthrough in the availability of knowledge engineering technology for practical, commercial use.

Architecture

M.1 provides facilities for knowledge engineers – people who design and implement expert systems – and for users who consult with expert systems to solve a particular problem.

Using any available text editor, a knowledge engineer creates a text file containing facts and rules in a special English-like language. When loaded into M.1, this file becomes the knowledge base M.1 uses in performing a consultation. As M.1 pursues a consultation, it interprets this knowledge base, storing interim conclusions in the cache.

The knowledge engineer can trace the flow of inferences and conclusions. Additional commands allow the knowledge engineer to add or remove entries from the loaded knowledge base, find values, view a list of entries containing a given expression, set and reset expressions in the cache, and continue with the consultation.

The knowledge engineer can also create a text file containing a "test case" of facts, to facilitate the debugging process.

After loading a knowledge base, the user consults with M.1 and receives advice on a specific subject. M.1 asks the user questions as it makes inferences based on the contents of the knowledge base. The user then supplies the required information or can in turn ask M.1 to explain why a piece of information is needed. The user is free to list knowledge base entries related to a particular aspect of the consultation, as well as to show the interim conclusions M.1 has noted.

M.1 also provides for saving and loading the cache with specific information from a disk file. This feature makes it possible to save a set of facts and conclusions for later use in a consultation or for interfacing with other programs.

M.1 – Operating Principles

M.1 reasons about symbolic expressions that denote aspects of some situation, such as the characteristics of a problem, its cause or its solution. M.1's reasoning is controlled by a knowledge base of facts, rules and meta-facts about the problem domain.

M.1 reasons about expressions which are English-like in appearance:
the weight of the drill bit

the recommended treatment for the patient.

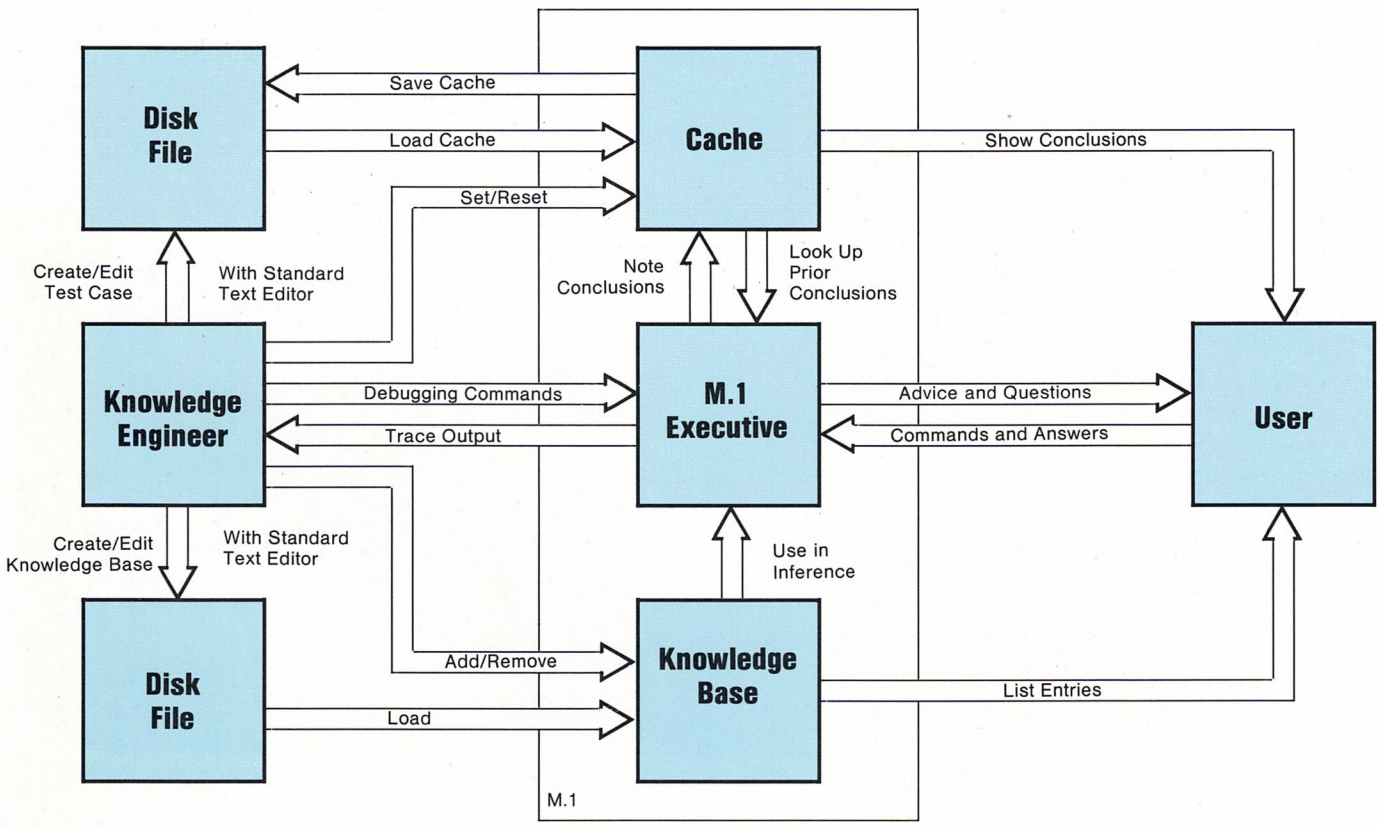
or more mathematical:
2 + 2
root-mean-square ([35, 30, 33, 35]) integral (2 • x / [log(x)– 1]).

Expressions have values associated with them, these values themselves being symbolic structures. M.1's basic operation is to find or to accumulate evidence for the value of expressions.

A consultation with M.1 typically consists of seeking a value for some special expression – the goal of the consultation – and then printing it. Unlike a conventional programming language, expressions in M.1 need not have just one value, nor does a value have to be certain. Typically, there will be evidence for several competing values.

A value is found in one of four ways:

- The knowledge base, or a data file read during a consultation, may contain a fact simply stating the value, for example: **age of patient = 34 years.**
- M.1 may infer a value by checking the truth of the premise of a rule whose conclusion states a value, for example:
  if distance-to-subject = N feet
  and N < 12
  and using-flash
  then shutter-speed = 125.



- M.1 may use a 'meta-fact' when finding a value. The most commonly used meta-facts state questions that M.1 may ask the user for the value of some expression, for example: **question (distance-to-subject) = 'How far from the camera is the subject of the photo?'.**
- M.1 may simply compute a value, as it does for arithmetic expressions and some other 'built in' functions.

This transparency of reasoning is exploited by M.1's explanation and justification system.

M.1 can justify the value it has found for some expression and during a consultation, explain its current line of reasoning. Combined with M.1's ability to reason with uncertain or incomplete knowledge, this transparency makes M.1 an ideal tool for building expert systems, which use 'rules of thumb' to mimic the reasoning process of a human expert.

M.1 – Knowledge Engineer Interface

English-like knowledge representation language.

This easy-to-read representation of facts and rules reduces the time and effort required to develop a knowledge base.

Powerful interactive debugging environment.

This includes an inference tracing mechanism, an interactive knowledge base "patcher" for making temporary changes to the knowledge base, a knowledge base auditor that keeps track of inferences and conclusions for later examination, and a knowledge base browser that can list all rules and facts that contain particular expressions. This environment greatly facilitates the expert system debugging process.

Text file format for knowledge bases.

This allows knowledge bases to be prepared by a knowledge engineer using a standard text editor.

Color display.

Colors are used to distinguish different types of interactions, such as questions, responses, commands, and error messages.

Screen oriented interface.

This "instrument panel" window-based interface allows the knowledge engineer to see the internal reasoning process of the system as it occurs.

M.1 – User Interface

Consultations in English.

The consultations takes place through questions and answers, in English. Therefore, the user can run a consultation with no special training or instruction.

Explanation facility.

During and after a consultation, the user can ask the system why a question is being asked, what the current conclusions of the system are, and how the current conclusions were established.

Help facility.

If the user is uncertain as to what response the system is expecting, typing "help" will provide guidance.

Automatic recognition of abbreviated answers.

The enduser need not type responses in full. Only enough characters to unambiguously distinguish the answer from the other possible answers are required.

Unknown, multiple, and uncertain answers.

If the answer to a question is not known, the user can type "unknown" rather than answering the question. The system will then proceed without the requested information. Similarly, the user can specify more than one answer to a question, if appropriate. If the answer to a question is uncertain, the user can attach a "confidence factor" to the answer, in the form of a percentage.

M.1 – Technical features

Uncertain knowledge.

The knowledge representation allows the encoding of uncertain knowledge through the association of certainty factors with facts and rules. These are automatically used during system execution to control the inference process and qualify conclusions. This allows the representation and use of heuristic, judgmental knowledge, as well as certain knowledge.

Automatic generation of questions.

If some information is required from the user during a consultation for which no prepared question has been entered by the knowledge engineer, an appropriate question will be automatically generated by the system.

Legal value checking.

User responses are checked for validity, if the knowledge engineer requests such checking. The system can check responses against a list of options or a range of numerical values, thereby explicitly screening out improper responses.

Arithmetic facilities.

The system handles both real and integer numbers, and provides built-in basic arithmetic functions and predicates.

List processing.

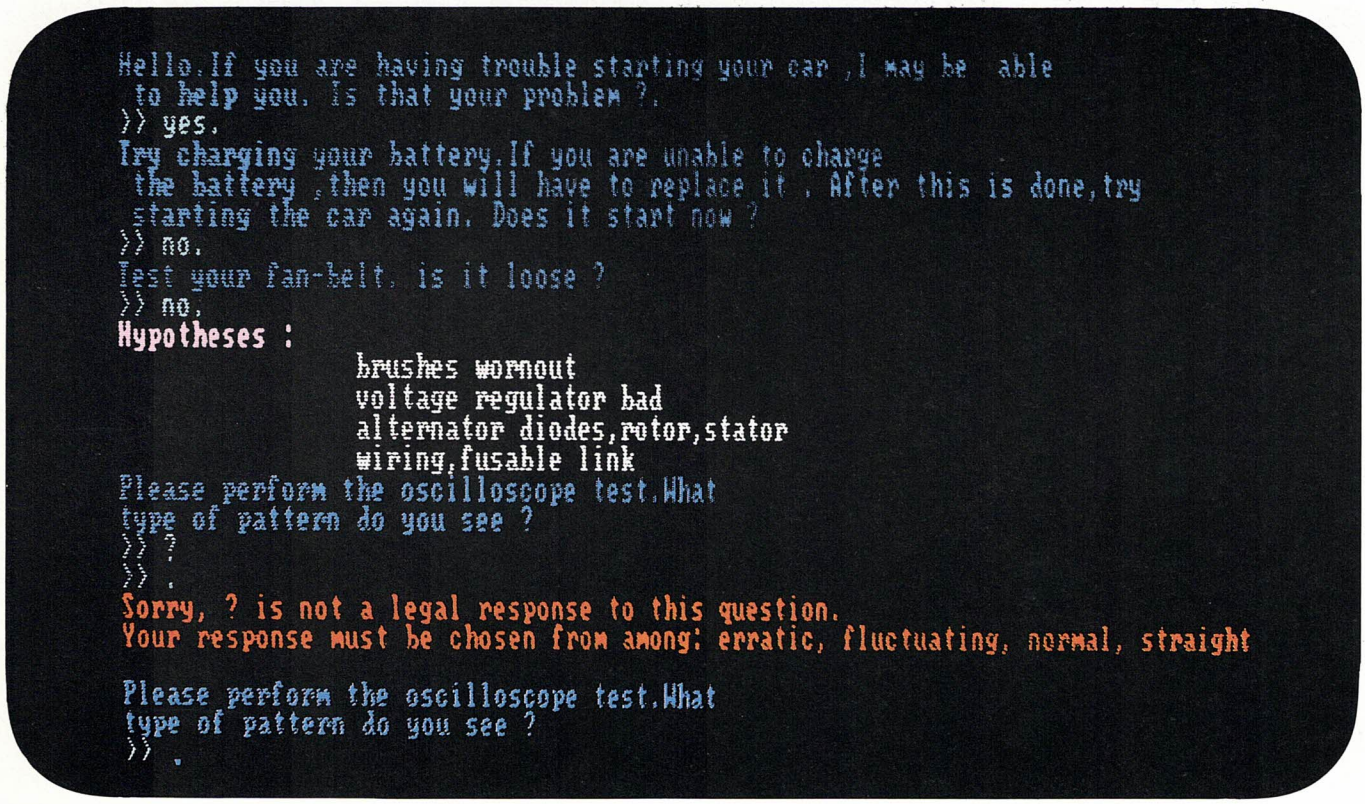
The system can reason about lists and other data structures, thus making it a complete symbolic programming language.

Multi-valued attributes.

The system supports multi-valued as well as single-valued attributes. For example, a company may have many vice presidents, but only one president; M.1 allows this distinction to be properly represented.

Variables in the knowledge base.

The knowledge representation language provides for the use of substitution



This screen illustrates a simple consultation with M.1 as it appears to a user. M.1 asks the user a series of questions (dark blue) and the user answers those questions (light blue). In this example, M.1 advises a user about car problems.

variables, which automatically take on values through pattern matching during a consultation. This powerful feature allows significant economies to be achieved in the construction of a knowledge base, when similar facts and rules can be collapsed into more general expressions.

Meta-propositions.

In building expert systems, it is often useful to alter the inference process depending on whether specific facts are known or unknown, are definite, or have other "meta" properties. Special built-in predicates allow this information about the state of knowledge of the system to be conveniently tested. In addition, the knowledge engineer can supply meta-facts in the knowledge base specifying, for example, the legal values of expressions.

Disk I/O.

The system can load knowledge bases dynamically during a consultation, providing an overlay capability for larger knowledge bases that can be segmented. In addition, the system can read and write disk files of facts and conclusions during consultations. This facility can be used to store test cases and to eliminate needless questions from consultations when information is available in machine readable form.

Conclusion monitoring facility.

When conclusions matching a specified pattern are made, a special set of high priority goals can be established, thereby enabling a limited forward chaining capability.

Presupposition checking.

The knowledge engineering can specify arbitrary presuppositions for goals. The system will attempt to establish the validity of these before pursuing the goal. This permits more succinct rules and additional control of the consultation.

Multiple objects.

The system contains built-in propositions that facilitate reasoning about multiple objects and collections of facts.

Hardware and Software Requirements

M.1 software operates on the IBM is a trademark of IBM Corporation Personal Computer under PC DOS 2.0 operating system and all IBM compatible PC's (Olivetti, ...). M.1 requires a minimum memory of 256K bytes. A color monitor and color board are recommended.

Selecting Applications

M.1 is best suited to a particular class of knowledge engineering problems known as "structured selection problems".

Structured selection problems are characterized by a finite set of possible solutions, and the use of a structured reasoning process to identify one or more of the possible solutions as appropriate for a given case. The steps in the reasoning process typically involve collecting, aggregating and abstracting information about the case, relating this abstract description to a general class of solutions, and then selecting and refining one or more specific conclusions.

Examples of structured selection problems include identifying a machine part failure, diagnosing an illness, recommending an appropriate tool for some task and advising on the options offered by a complex computer program.

Among the cues used to find an appropriate application for M.1 are:

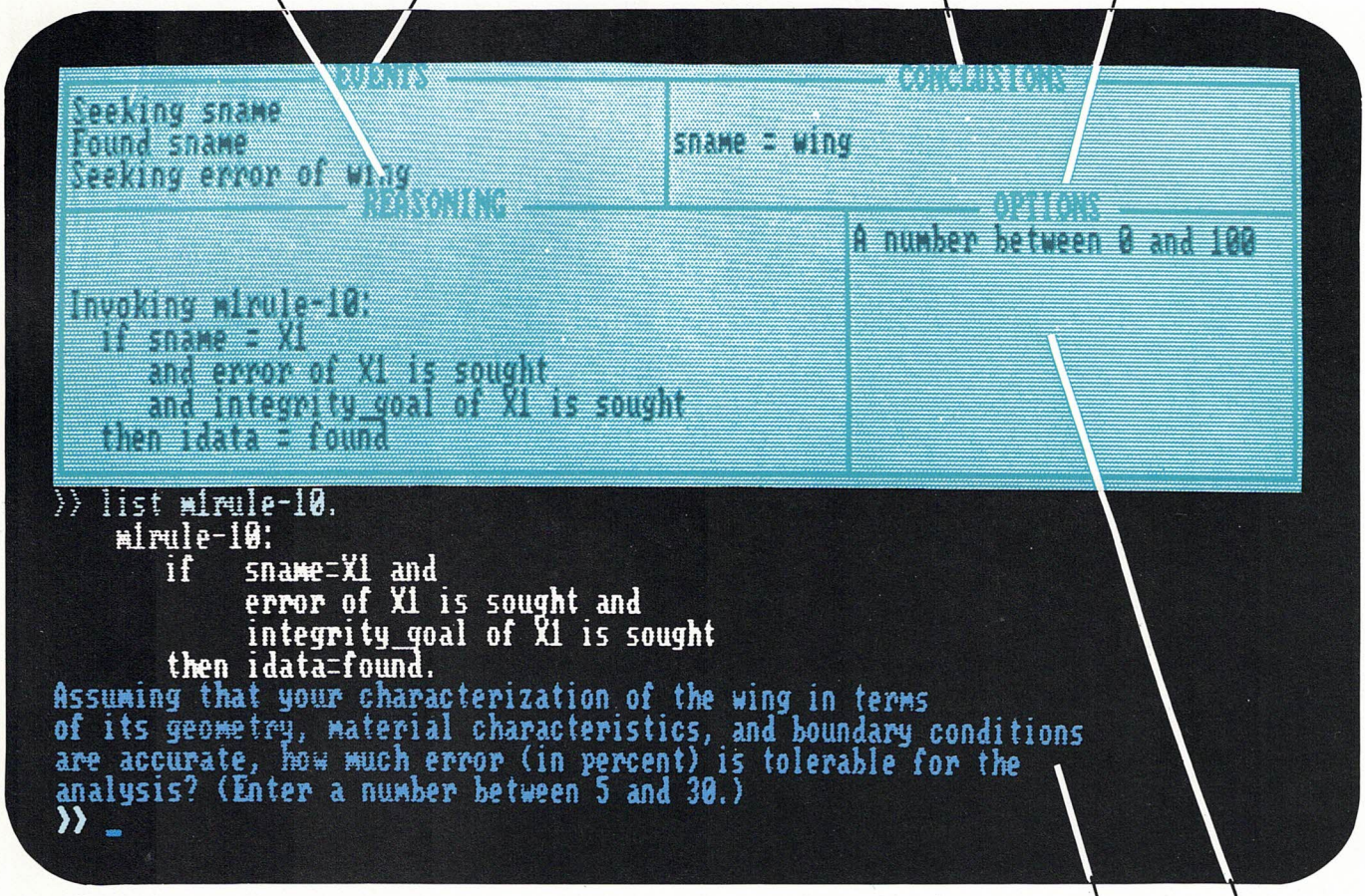
- There is a human expert who can usually solve the problem in a thirty minute consultation.
- No diagrams, nor direct physical contact are needed to solve the problem, i.e. the problem could be solved in a phone conversation.
- The solution involves informal reasoning. The problem does not require extensive calculation or formal analysis.
- There are at most a few dozen possible conclusions.

The REASONING window displays knowledge base entries as they are considered and indicates whether or not they are applicable in the current consultation.

The CONCLUSIONS window contains intermediate or final conclusions reached by the system during a consultation. These conclusions may be certain, or may be modified by a "certainty factor" on a scale from 1 to 100.

The EVENTS window indicates significant events in the reasoning process, such as seeking the value of an expression.

The OPTIONS window displays the options available to the user in responding to the current question. Alternatives include lists of possible answers and numerical ranges.



With the "instrument panel" on, a knowledge engineer can examine the internal operation of the system as it occurs. This aids in developing the knowledge base and debugging the reasoning. The upper half of the screen is devoted to windows that display the current state of the deduction. The lower half of the screen shows the consultation as it appears to a user of the completed knowledge system.

M.1

The Wine Advisor

An expert considers several factors when selecting a wine: first, there are the characteristics of the meal that the wine will be served with, then the preferences of the person who will be drinking the wine, and the relative qualities of the wines themselves. Typically, an expert will select a best choice based on the meal itself. As the expert gathers more information about the meal and the diner's preferences, this opinion is confirmed or reformed.

The Wine Advisor, an example system with approximately 100 knowledge base entries, follows the deliberation process of a typical California wine expert.

```
Is the main component of the meal meat, fish or poultry?
>> poultry.

Does the meal have turkey in it?
>> no.

Does the meal have a sauce on it?
>> yes.

Is the sauce for the meal spicy, sweet, cream or tomato?
>> tomato.

Is the flavor of the meal delicate, average or strong?
>> average.

Do you generally prefer dry, medium or sweet wines?
>> dry.

The following wines will mostly be dry, medium-bodied, and red.
They are recommended for your meal.
    the wine for cycle 1 = zinfandel (80%) because rule-27.
    the wine for cycle 1 = cabernet sauvignon (80%) because rule-27.
    the wine for cycle 1 = burgundy (30%) because rule-27.
    the wine for cycle 1 = valpolicella (30%) because rule-27.
```

```
Are you happy with these?
>> no.

Which characteristic of the wine would you like to change?
>> color and sweetness.

What color would you prefer?
>> white.

What sweetness would you prefer?
>> medium.

The following wines will mostly be medium, medium-bodied, and white.
They are recommended for your meal.
    the wine for cycle 2 = chardonnay (72%) because rule-27
    the wine for cycle 2 = riesling (72%) because rule-27.
    the wine for cycle 2 = chenin blanc (30%) because rule-27.
    the wine for cycle 2 = soave (30%) because rule-27.

Are you happy with these?
>> yes.
```

```
        C:SACON  FC=3774 FL=139 COL 01
'Solution will be based on a modified Newton-Raphson approach.',nl,
'  Perform incremental stress and incremental strain analyses.',nl,nl,
'Model nonlinear stress-strain relation of the material.'].

rule-43:
        if stress of STRUCTURE = fatigue
        then analysis_recs of STRUCTURE = [nl,
'Special code should be written to scan peak stress at each step,',nl,
'  and evaluate fatigue.', nl,nl,
'A single cycle of loading is sufficient for fatigue estimates.'].

rule-46:
        if stress of STRUCTURE = stress_exceedence
        then analysis_recs of STRUCTURE = [nl,
'Special code should be written to scan stresses, smooth stresses,',nl,
'  and compare with allowable stresses (with appropriate safety factors).'].

rule-47: if deflection of STRUCTURE = excessive_deflection
        then analysis_recs of STRUCTURE = [nl,
'Special code should be written to scan deflections, calculate',nl,
'  relative values, and compare with building code limits.'].

rule-51:
1LFTEND 2RHTEND 3DELCHR 4DELWRD 5UNDLIN 6RHTWRD 7BEGBLK 8ENDBLK 9BEGFIL 10ENDFIL
```

During or after a consultation, the user or knowledge engineer can display portions of the knowledge base and conclusion cache, or request an explanation of the reason why the current question is being asked.

M.1 – The Structural Analysis Consultant (SACON)

A version of this classic expert system, developed at Stanford University, has been recoded into M.1, as an example of the power and flexibility of the system.

It contains over 200 knowledge base entries. The M.1 version is comparable in response time to the original, which ran on a large mainframe computer. SACON helps the user in determining the sequence of analysis steps and advises on potential problems should be alert to.

In this example, the engineer is concerned with the analysis of a 747 aircraft wing.

```
What is the name of the structure you wish to analyze?
>> 747-wing.

Assuming that your characterization of the 747-wing in terms
of its geometry, material characteristics, and boundary conditions
are accurate, how much error (in percent) is tolerable for the
analysis? (Enter a number between 5 and 30).
>> 5.0.

Do you want to examine the responses of the 747-wing,
evaluate its instability, or both?
    *examine.
    *evaluate.
    *examine and evaluate.
>>examine.

Does the 747-wing have any time dependent terms in its
equations of equilibrium?
>> no.

Is heating or cooling of interest in determining the structural
responses of the 747-wing?
>> no.
```

```
What are the names of the substructures of the 747-wing?
(Enter a single name if there is only one substructure. If there is
more than one, enter the names separated by commas.)
>> total-wing.

What is the material composing most of the total-wing?
>> options.

Your response must be chosen from the following:
high-strength-aluminum, upgraded-aluminum, structual-aluminum,
high-strength-steel, upgraded-steel, structural-steel, c
What is the material composing most of the total-wing?
>> high-strength-aluminum.

Enter the average longest dimension of the total-wing in inches.
(Use the format: xxx.x.)
>> 1420.0.

Enter the weight of the total-wing in pounds.
(Use the format: xxx.x.)
>> 46000.0.

Is the total-wing thin-walled or solid?
>> thin-walled.
```

```
What is the configuration of support for the total-wing?
>> one-side.

Is the total-wing to be modelled as a 1, 2, or 3 dimensional
system?
>> why.

M.1 is trying to determine whether the following rule is applicable
in this consultation:

rule-54:
    if modelling dimentionality (total-wing) = 1 and
    construction (total-wing) = network
    then shape (total-wing) = beam.

The following knowledge base entries are also being used:
    metafact-4          (a whenfound)
    rule-52             (a rule)
    rule-10             (a rule)
    rule-22             (a rule)
    rule-25             (a rule)
    kb-1                (a initialdata)

Is the total-wing to be modelled as a 1, 2, or 3 dimensional
system?
>> 2.
```

```
Would you describe the construction of the total-wing to be a
    * network.      (truss and/or frame elements)
    * continuum.   (sheets and/or blocks of material), or
    * network and continuum. (a combination)?
>> continuum.

Is the overall geometry of the total-wing curved or planar?
>> curved.

What are the names of the loadings on the total-wing?
(Enter a name, or names separated by commas.)
>> flight.

Enter the number of cycles of flight that will be applied.
(Use the format: xxx.x.)
>> 20000.0.

How many components are there to flight?
>> 1.

Would you describe flight-component-1 as being distributed over
most of the substructure or as acting at a point of the substructure?
    * distributed.
    * point.
>> distributed.
```

```
Which surface of the substructure does flight-component-1 act
normal to?
>> width-length.

Enter the average wall thickness of the total-wing in inches.
(Use the format: xxx.x.)
>> 0.31.

Enter the average depth of the total-wing in inches.
(Use the format: xxx.x.)
>> 41.0.

Enter the magnitude of the distributed load (in psi).
(Use the format: xxx.x.)
>> 1.0.

Do the supports of the 747-wing involve Coulomb friction,
nonlinear springs, and/or gapping? (Answer yes or no.)
>> no.

The following analysis classes are relevant to the analysis of
your structure:

general inelastic
```

```
The following are specific analysis recommendations
you should follow when performing the structure
analysis:

Activate incremental stress-incremental strain analysis.

Model nonlinear stress-strain relations of the material.

Solution will be based on a mix of gradient and Newton methods.

Special code should be written to scan peak stress at each step,
and evaluate fatigue.

A single cycle of loading is sufficient for fatigue estimates.

Special code should be written to scan stresses, smooth stresses,
and compare with allowable stresses (with appropriate safety factors).

Special code should be written to scan deflections, calculate
relative values, and compare with engineering code limits.

Cumulative strain damage should be calculated.

Analysis should include two or more load cycles (if cyclic)
with extrapolation for strain accumulation.

Special code should be written to perform shakedown extrapolation.
```

M.1

The Bank Services Advisor

The Bank Services Advisor is a small expert system, containing approximately 200 knowledge base entries. It provides advice to customers about the services of a bank.

In establishing a banking relationship, customers are typically faced with selecting from a bewildering array of services, some more appropriate for their needs than others. In most banks, this information is available only from numerous pamphlets or from bank representatives who may have an imperfect knowledge of the changing and complex services available. Dispensing such information through an expert system could provide timely, accurate, and confidential financial advice on an individual basis. The bank could increase service, reduce costs, and improve its competitive position.

***** Welcome to the Bank Services Advisor *****

This new intelligent computer system will help you to select the services of our bank that are most appropriate for you. To determine the services that best fit your needs, the system will ask you a series of questions to develop a profile of your banking requirements. It will then make recommendations about how you can most effectively utilize the many services of our bank.

Please type your answers to the questions on the keyboard located below the screen. If you are unsure of the answer to a particular question, simply type "unknown". Enter all dollar amounts to two decimal places. Be sure to end all of your responses with a period.
Are you generally familiar with the services our bank provides?
>> no.

Thank you for giving us the opportunity to explain our special services. You will see a brief description of each service before you are asked if you want the service.

We offer many different styles of checks and check books to suit your check writing needs.
Do you want to be able to write checks?
>> yes.

One of our most convenient services is automatic deposit – of your paycheck, Social Security or other government checks, or other regularly-paid income such as annuities.

Do you want automatic deposit services?
>> yes.

One of our popular services is automatic transfer of funds to pay bills which are due at the same time and in the same amount every month – loan payments, for example.

Do you want automatic transfer service?
>> yes.

Our bank offers a new phone transfer service which allows you to pay bills by merely punching buttons on your touch-tone phone. You can pay bills for most utilities, most major credit cards, telephone service, many department and chain stores, many insurance carriers, and others. There are no special fees for this service; thus, you will save time and postage. As a bonus, all of our auto-teller services are available to you.

Do you want our new phone transfer service?
>> yes.

How much will you deposit initially?
>> 10000.0.

How many deposits do you expect to make each month (including any automatic deposits)?
>> 2.

What is the average amount that will be deposited?
>> 2000.0.

How many cash withdrawals do you expect to make at the bank each month (not including any auto-teller withdrawals)?
>> 0.

How many cash withdrawals do you expect to make at an automatic teller each month?
>> 1.

What is the average amount of each auto-teller withdrawal?
>> 100.0.

How many automatic transfers (e.g., to pay loan accounts) do you expect to make each month?
>> 5.

What is the average amount of each automatic transfer?
>> 100.0.

How many checks will you be writing each month?
>> 15.

What is the average amount of each check?
>> 50.0.

In the worst case, your balance during the month could become as low as $ 8,650.00.
Do you approve this worst case minimum balance? (If you say no, you will be asked to enter a different figure for minimum balance.)
>> no.

Practically speaking, what do you think your minimum balance will be?
>> 10000.0.

A combination of checking and savings accounts may satisfy your needs best. Do you want such a combination of accounts?
>> yes.

[Considering best savings account...]

A time deposit account may be appropriate for you. Such an account requires you to leave your initial deposit in the account for a specified time period, usually some months long (ranging from one week to five years). You may withdraw interest when it is paid (usually monthly or quarterly), or leave it to be reinvested, but you may not withdraw any of the principal (initial deposit) without severe interest penalties.

Do you want a time deposit account?
>> yes.

What is the longest time period you would consider for a time deposit account? In other words, what is the longest time you would be able to leave your money in the account without withdrawals, except for interest? (Please specify a number followed by: days, weeks, months, or years).
>> 6 months.

[Considering best checking account...]

Recommended account: time deposit (2)
This account is a time deposit account with a term of seven days.

If you can limit yourself to depositing and withdrawing funds only on a particular day of the week (e.g., Monday), and only in person at the bank or by mail, this is a good choice. If you need more flexibility, try one of the interest-bearing checking accounts.

Recommended account: money market checking.
This account is an interest-bearing checking account. It has a higher rate of interest than our regular interest checking account and requires a higher minimum balance. Unlike our interest-bearing limited checking account, you may write an unlimited number of checks.

M.1 – Course

The four-day M.1 course teaches the operation, features, and use of the system. The course is for programmers. No prior experience in artificial intelligence or knowledge engineering is required.

The M.1 training course is an intensive hands-on seminar. Participant have use of IBM Personal Computers throughout the four days. Lectures, discussions, and videotapes cover the technical features of the tool and their relevance to the practical problems of building expert systems. Scripts, guide exercises and small projects follow up on the points made in the lectures.

The course covers the following topics:

- What the technology does – The basic operation of M.1, and the problems it is suited for.
- How the technology works – The basic mechanism underlying M.1, including the role of backchaining, the order of entries in the knowledge base, and the distinction between facts and meta-facts.
- How to represent knowledge – Advanced knowledge representation features of M.1.
- How to control consultations – Methods for tailoring and controlling the inference process in the course of a consultation, including iterative data gathering cycles, hypothesize-and-test structures, and writing numerical and other procedures.
- How to build a substantial application – Experiences of the original development team in building SACON, a classic expert system: Problems encoutered and how these problems were solved. Includes a walkthrough of the SACON knowledge base and a sample consultation.
- Knowledge engineering methodology – How to select projects, design knowledge bases, and plan and schedule a knowledge engineering project.