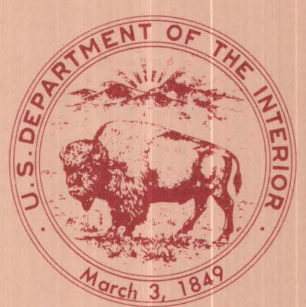
The muPROSPECTOR Mineral Consultant System

U.S. GEOLOGICAL SURVEY BULLETIN 1697



The muPROSPECTOR Mineral Consultant System

By RICHARD B. McCAMMON

A microcomputer-based
expert system for regional
mineral resources
evaluation

U.S. GEOLOGICAL SURVEY BULLETIN 1697

DEPARTMENT OF THE INTERIOR
DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY
Dallas L. Peck, Director

UNITED STATES GOVERNMENT PRINTING OFFICE: 1986

For sale by the Books and Open-File Reports Section, U.S. Geological Survey,
Federal Center, Box 25425, Denver, CO 80225

Library of Congress Cataloging-in-Publication Data
McCammon, Richard B.
The muPROSPECTOR mineral consultant system.
(U.S. Geological Survey bulletin ; 1697)
Bibliography: p. 6
Supt. of Docs. no.: I 19.3:1697
1. MuPROSPECTOR (Computer system). 2. Mines and
mineral resources—New England—Data processing.
I. Title. II. Series.
QE75.B9 no. 1697 86-600276 557.3s [TN153] [553'.028'55369]

CONTENTS

Abstract 1
Introduction 1
Overview of muPROSPECTOR 1
muPROSPECTOR commands 2
Running muPROSPECTOR 2
Building models 3
    Getting started 3
    Representation of models 3
    Model file format 4
Programming aids in model-building 5
    Global variables 6
    Functions 6
The muPROSPECTOR program 6
Uses and limitations 6
References cited 6
Appendix A.—Session with muPROSPECTOR 10
Appendix B.—Backus-Naur Form (BNF) description of the formal language for external representation of muPROSPECTOR models 17
Appendix C.—Program listing of muPROSPECTOR 19

FIGURES

1. A simple inference (I) from evidence (E) 4
2. A simple inference net 5

The muPROSPECTOR Mineral Consultant System

By Richard B. McCammon

Abstract

The muPROSPECTOR mineral consultant system is a computer program to aid geologists in evaluating the potential mineral resources in an area. External to the program is a knowledge base that consists of mineral deposit models. In a consultation, the answers given by the geologist to questions asked by the program are used to estimate the likelihood of occurrence of the deposit types represented in the knowledge base. The current knowledge base contains models for seven major deposit types; many more such models are needed for the future.

INTRODUCTION

muPROSPECTOR is a computer program intended to aid the geologist in evaluating the potential mineral resources of an area. External to the program is a knowledge base that consists of models of mineral deposit types. Models are based upon the expertise and judgment of geologists familiar with the settings of known deposits. The current knowledge base comprises deposit models appropriate to the major tectonostratigraphic terranes in New England. Included are models for volcanogenic and sediment-hosted massive sulfide deposits, porphyry copper deposits, uranium in vein and in young organic-sediment deposits, tin greisen deposits, and modern placers. Computer listings of these models can be found in an open-file report by McCammon and others (1984). They were used to aid in the assessment of the potential mineral resources in the Sherbrooke and Lewiston 1°×2° quadrangles, Maine, New Hampshire, and Vermont, as part of the Conterminous United States Mineral Assessment Program. The current version of the program has been implemented on the IBM PC[1] and is written in the muLISP programming language (The Soft Warehouse, 1985). It is patterned after the PROSPECTOR system developed at SRI International (Duda, 1980).

This paper is intended for geologists who wish either to use the existing models in muPROSPECTOR or to build new models. The program is written so that new models can be incorporated within the existing knowledge base. Models can be built for other mineral deposit types or for other kinds of analyses, such as sedimentary basin models for petroleum resource evaluation.

OVERVIEW OF muPROSPECTOR

Deposit models are represented in muPROSPECTOR by inference networks that relate field observations to attributes of models. An inference network is equivalent to a collection of inference rules. In muPROSPECTOR, an inference rule has the form

IF E THEN H ELSE NOT H.

The rule is interpreted to mean "The observed evidence E either confirms or rejects the hypothesis H." This is a simplification of the more powerful rule in PROSPECTOR that has the form

IF E THEN (to a certain degree) H.

(Reboh, 1981). Different pieces of evidence can be combined logically to form a single, compound piece of evidence. The simpler elements are combined by means of the logical operations of conjunction (AND), disjunction (OR), and complementation (NOT). Rules interconnect through "chains" where the hypothesis for one rule is the negative evidence for another, through several pieces of evidence bearing on the same hypothesis, and through the same piece of evidence bearing on several different hypotheses.

During a consultation, the user is expected to answer YES or NO (Y or N) to each question. The questions relate to the various pieces of evidence that may be present in the area. Each question is preceded by a sequence number that identifies the question (and its corresponding answer) and can be used later to reference a question asked earlier in a consultation.

[1] Use of trade names in this publication is for identification purposes only and does not imply endorsement by the U.S. Geological Survey.

In the PROSPECTOR system, answers to questions are numbers between −5 and +5, which indicate the certainty of the user's response (Duda, 1981): a certainty of −5 is equivalent to "N" in muPROSPECTOR, and a certainty of +5 is equivalent to "Y" in muPROSPECTOR. A value of "0" in PROSPECTOR indicates that the user has no opinion one way or the other. In muPROSPECTOR, the user answers "D" (default) to such a question. In this case, the program puts the user in another part of the inference net to try and answer the question. If there is no alternative, the answer to the question is recorded as uncertain but is treated as if the answer were "N." The reason for this treatment is that for most deposit models, it is positive evidence that is being sought. A "don't know" response, if no alternatives are specified, produces the same result as if the evidence were absent.

In addition to answering questions "Y", "N", or "D" (don't know), the user can be asked to select a number of choices for a given question, as shown by the following example taken from the Ordovician-Carboniferous plutonic terrane model:

1—For which of the following do you have any information:

1) Age of intrusive is Ordovician to Carboniferous (500–290 m.y.)
2) Age of intrusive is Ordovician to Silurian (500–410 m.y.)
3) Age of intrusive is Devonian (410–360 m.y.)
4) Age of intrusive is Carboniferous (360–290 m.y.)

Please enter one or more of the preceding numbers (separated by blanks and terminating with a ⟨CR⟩)

Selecting more than one choice enables the user to pursue multiple lines of evidence and permits multiple inferences for a set of observations. For instance, more than one type of deposit may be present in the area under study. Multiple choices make it possible to examine a large number of deposit types in relation to the study area.

muPROSPECTOR COMMANDS

During a consultation a user may type any one of the muPROSPECTOR commands described below in order to complete the task at hand.

These commands are

A(CCESS)

By typing "A", the user can access a database in order to answer a question. muPROSPECTOR is temporarily suspended while the particular database is being accessed. Upon completion, control is returned and muPROSPECTOR picks up where it left off.

S(UMMARIZE)

By typing "S", the user can have muPROSPECTOR summarize its conclusions on the basis of the information provided. A summary consists of the factors of models that have been established, the factors that are uncertain, and the conclusions that have been made. The factors are listed in the order in which they were established, making it possible for the user to assess the value of the information provided at each stage of the consultation.

T(RACE)

By typing "T", the user can see the intermediate conclusions made by muPROSPECTOR after each question is answered. This command allows the user to monitor the progress of the consultation.

U(NTRACE)

By typing "U", the user stops the trace.

(SHO)W

By typing "W", the user can examine answers to previous questions or earlier choices made. The user is requested to give the number (or numbers) of the previous questions to be examined. muPROSPECTOR displays the question (or list of choices) and the answer (or choices made).

?

By typing "?", the user can request muPROSPECTOR to display either a more detailed rephrasing of the question or an explanation of why the question is being asked. This command is useful when the user does not understand the question or wishes to know the background and justification for the question being asked. A rephrasing or an explanation may not always be available. In such cases, muPROSPECTOR informs the user of this fact and repeats the question.

B(REAK)

By typing "B", the user interrupts muPROSPECTOR and enters the BREAK condition in muLISP. This command suspends muPROSPECTOR and allows the user to evaluate S-expressions (symbolic expressions; the data objects to be manipulated) in muLISP. It permits such operations as inspecting the variables in muPROSPECTOR, evaluating functions defined in muPROSPECTOR, or using the EXECUTE command to execute programs outside of muPROSPECTOR. To return to the consultation, the user types (RETURN). The BREAK command is intended for use by those familiar with muLISP.

Q(UIT)

By typing "Q", the user exits muPROSPECTOR and returns to the system.

RUNNING muPROSPECTOR

A transcript from a sample consultation with muPROSPECTOR is given in Appendix A. It is designed to illustrate many of the features of the program. In the transcript, the responses appear in bold type immediately

below each question. The numbers enclosed within parentheses on the right side were inserted after the session was terminated and are keyed to the numbered comments following the transcript.

The basic consultation has three phases: selection of a model (or models), matching of a model (or models), and evaluation. Each of these three phases is shown in the transcript.

BUILDING MODELS

The design of new models for muPROSPECTOR is a creative and challenging activity, involving the synthesis of scientific understanding about types of deposits into a precise language. The refinement of new models requires iteration as each model is tested against known deposits and adjusted accordingly. For the PROSPECTOR system, the design and debugging of new models are discussed in detail by Duda and others (1978). In muPROSPECTOR, the system is less complex and requires fewer iterations.

Getting Started

In order to create a new muPROSPECTOR family of models, two files with extensions ⟨*.HDR⟩ and [⟨*.MDL⟩] must be created. (The notation used for file names and the syntax of arguments are described in Appendix B.) The family of models is defined by the file labeled ⟨Filename⟩.HDR. This file contains the following information stored in a keyword/value format in which each value is delimited by "/*" and "*/" as follows:

```
TITLE /* ⟨text⟩ */
HELLO /* ⟨text ⟩ */
PURPOSE /* ⟨text⟩ */
FILENAMES /* [⟨models⟩] */
OBJLST /* [⟨model-names⟩] */
BANNER /* ⟨banner⟩ */
OBJECT /* ⟨object⟩ */
ATTRIBUTE /* ⟨attribute⟩ */
STOP
```

The following example is from NERAP.HDR (the file name of the New England Mineral Resource Assessment Program):

```
TITLE /* * * NEW ENGLAND MINERAL RESOURCE ASSESSMENT PROGRAM * * */
HELLO /* I am here to assist you in assessing the potential for metallic deposits in New England */
PURPOSE /* This is a program to assist ... */FILENAMES /* GEN METAVOLC METASED OCPLUT JVLPL SURF */
OBJLST /* Unknown Metavolcanic Metasedimentary Ordovician-to-Carboniferous-plutonic Jurassic-volcanic-plutonic ... */
BANNER /* TERRANE CLASSIFICATION */
OBJECT /* terrane */
ATTRIBUTE /* area */
STOP
```

Representation of Models

For muPROSPECTOR to make use of models, each model must be stored as a file. Its form is different from ordinary prose or textbook descriptions. The model must be highly structured so that the program can make inferences by examining the parts of the model and the relations among them. Associated with each piece of evidence is an inference about a factor considered important with respect to a model. In muPROSPECTOR, the model information is stored as a collection of inference rules each having the form:

$$\text{IF } E_1 \text{ AND } E_2 \text{ AND } \ldots \text{ AND } E_n \text{ THEN I.}$$

The rule is interpreted to mean "The n pieces of observed evidence E_1 through E_n imply the inference I." If the rule mentions only a single piece of evidence E, we can represent the rule graphically as shown in figure 1. We refer to the rectangles as spaces and to the arrows connecting the spaces as pointers.

As a simple example, consider one rule in the model for Ordovician-Carboniferous plutonic terranes:

```
IF age of intrusive is Devonian (410–360 m.y.)
    intrusive best described as a pink to gray,
        variably porphyritic, pegmatitic or two-
        mica granite
    major faults in the vicinity or intrusive highly
        fractured
    potash-enriched zones identified in the intru-
        sive
THEN Possible potential resource for uranium in
    vein deposit.
```

This rule states that according to the model, observation of this evidence is encouraging for the occurrence of one or more uranium vein deposits in the area being evaluated. In effect, the favorable factors suggested by the evidence are that the intrusive is the right age, there is a favorable uranium source rock, there is evidence of fractures, and a potash-enriched zone is present.

In general, the rules in muPROSPECTOR connect in various ways—through chains, through several pieces of evidence bearing on the same general inference, and through one piece of evidence bearing on several different inferences. Thus, the collection of rules forms an inference network, such as the one shown in figure 2. The space at the top of the network is termed the topspace for a given model. Associated with each pointer is an answer to a question or a choice among a set of choices. Reasoning in muPROSPECTOR is from the top down within an

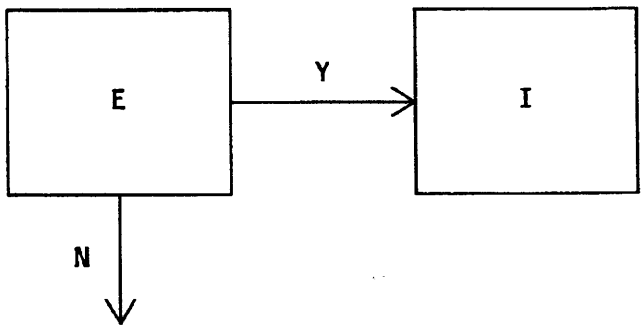


Figure 1. A simple inference (I) from evidence (E); Y, yes; N, no.

inference network. This is called forward-chaining. Multiple inferences in muPROSPECTOR are made possible by multiple choices incorporated within the network. It is the responsibility of the model-designer to assign the likelihoods associated with each rule. In most cases, the likelihood is expressed by a verbal description that reflects degree of certainty. Once the critical factors of a model are identified, the main task of the model-designer is to arrange the factors in a network that best reflects expert judgment based on different pieces of evidence.

Model File Format

As indicated above, each ⟨model⟩ is stored as a file with the name ⟨model⟩.MDL. Beginning each file is the following:

model ⟨model⟩
topspace ⟨space⟩

where ⟨model⟩ is the filename and ⟨space⟩ is the topspace of the model. A model file consists of spaces. Each space is defined by a set of keywords having values. Values that are represented by a ⟨list⟩ are delimited by " /*" and " */"; that is, /* ⟨list⟩ */. Values that are atoms (single words) have no delimiters. Only the keyword STOP has no value and is always the final entry in the file. The following describes the representation of the different types of spaces contained in a model file:

1) Spaces that represent a piece of evidence are of the form

space ⟨space⟩ desc ⟨inference⟩ ques ⟨question⟩ expl ⟨explanation⟩ yes ⟨space1⟩ no ⟨space2⟩ def ⟨space3⟩

where ⟨inference⟩ bears on the factor associated with the ⟨space⟩, ⟨question⟩ contains the question being asked, ⟨explanation⟩ contains the explanation if requested, ⟨space1⟩ refers to the space to be considered next if the answer to the question is yes, ⟨space2⟩ to the space if the answer is no, and ⟨space3⟩ to the space if the answer is "don't know." If def is missing, a "don't know" answer is taken as a "no" answer. If expl is missing, a default explanation is given if asked for by the user.

2) Spaces that represent choices are of the form

space ⟨space⟩ desc ⟨inference⟩ choices /*[⟨space⟩] */

where [⟨space⟩] refers to a list of the spaces the user is asked to choose from. Each space on the list must have in addition the following:

noun-phrase /* ⟨label⟩ */

where ⟨label⟩ is the description of the choice represented by the space.

3) Spaces that represent inferences are of the form

space ⟨space⟩ inf ⟨inference⟩

where ⟨inference⟩ is the inference based on the evidences represented by all of the connected spaces. A keyword that can appear in any space is

announce ⟨announcement⟩

where ⟨announcement⟩ is a statement that is printed out to the user before any action is taken for the space in which it is contained. In a complex model, this facilitates the consultation by providing additional information to the user.

For questions that can be answered by accessing some database, the keyword is

database /* ⟨command⟩ */

where ⟨command⟩ is the command that initiates a program to access a database. If no database is available for a space in which the user answers "A," a default message is printed indicating that a database to assist in answering the question is not available.

With one exception, spaces and their associated keywords with their values can be arranged in any order in the model file. In certain instances, different spaces will have the same value for particular keywords. Rather than repeating the values, the name of the space with the same value is given as the value of the keyword. When this is done, the space referred to must appear before the reference. As an example, consider the space ALPHA with keyword/value ques /* Do you observe faults in the vicinity */. In a different part of the model, there might be a space BETA with the same question, in which case space BETA would be represented as

space BETA ques ALPHA.

Space ALPHA and its associated keywords with their values appear before space BETA and its associated keywords with their values. In general, a space OMEGA could be represented as

space OMEGA ques ALPHA expl BETA announce GAMMA database DELTA

provided that the spaces ALPHA, BETA, GAMMA and DELTA appeared earlier in the file listing.

For the family of models [⟨model⟩] stored in ⟨family⟩.HDR, it may happen that answers to questions in ⟨model1⟩ may cause a switch to ⟨model2⟩. In such cases, the value of the appropriate keyword for the space in ⟨model1⟩ is given as ⟨space⟩.⟨model2⟩. For example, in the model GEN, for space ALPHA, if the answer to the question is "Y," control is switched to space BETA in

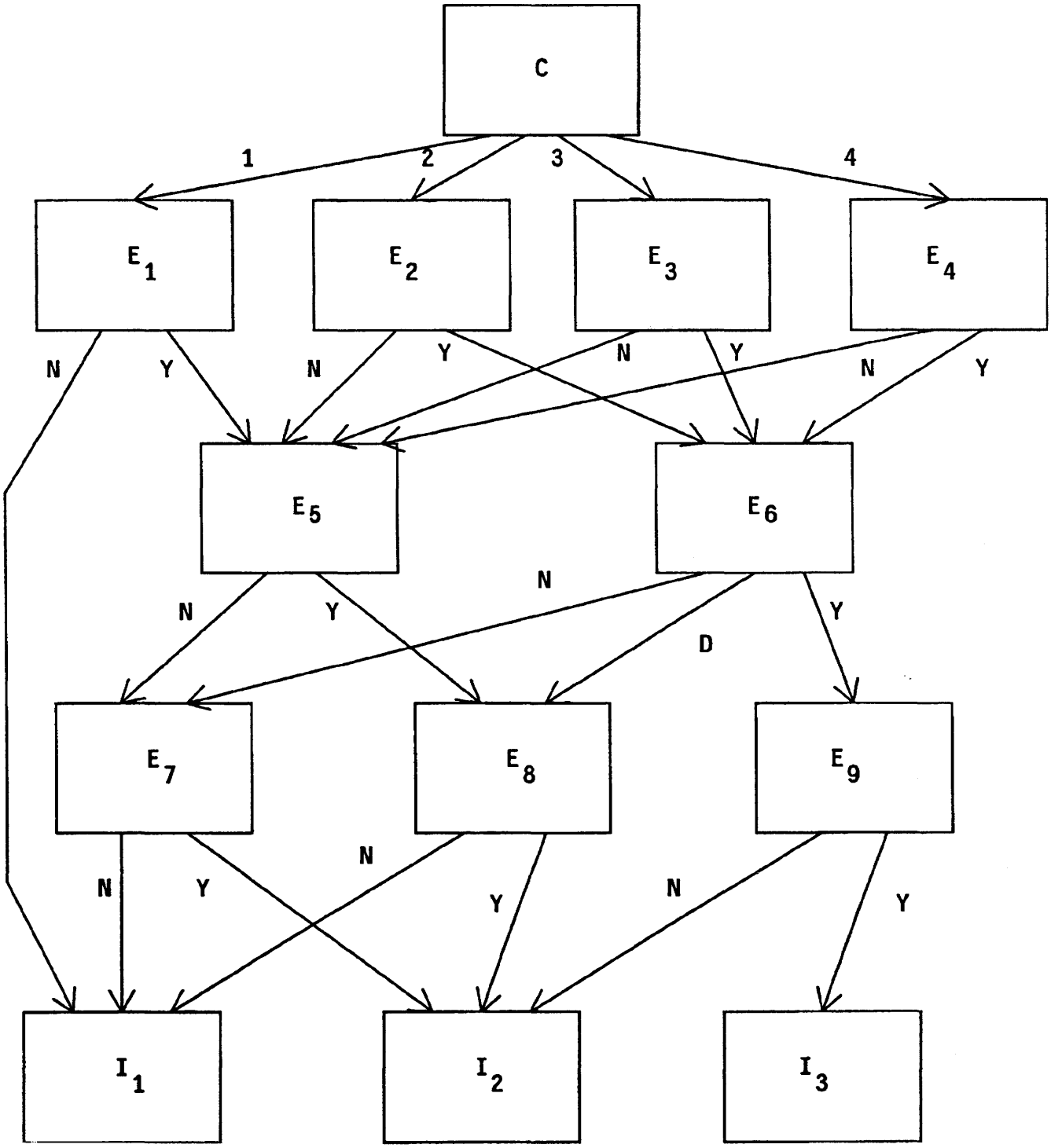


Figure 2. A simple inference net. C, multiple choice; E, evidence; Y, yes; N, no; D, don't know.

model METAVOLC. The value for the keyword yes in space ALPHA is BETA.METAVOLC. This naming convention makes it possible for models within a family to be interconnected. This ability is particularly useful when building a general model that contains pointers to other, more specialized models.

PROGRAMMING AIDS IN MODEL-BUILDING

It was mentioned earlier that a B(REAK) during a consultation in muPROSPECTOR puts the user in the muLISP environment. This command can be helpful both

when models are being created and when they are being tested. Entering a BREAK during a run permits the model-builder to inspect the values of global variables, to evaluate functions that return the contents of spaces within a model, and to examine the current state of the program.

Global Variables

The following is a list of muPROSPECTOR's global variables that may be useful in debugging new models. Their values may be inspected within a BREAK during a muPROSPECTOR run.

Variable	Description
RDS	Model currently active
FILENAMES	List of the models available
LOADED-MODELS	List of the models currently stored
MODEL-SPACES	List of the spaces currently stored
QUESTN	Number of questions that have been answered
RESULTS	List of the choices, answers, and inferences made up to this point in the consultation
SPACE	Space currently active (NOTE: The full name, that is, ⟨space⟩.⟨model⟩, is stored internally in the program)

Functions

The following table lists those muPROSPECTOR functions that might be useful to the model-builder during encoding of a model. These functions can be evaluated in a BREAK during a run of muPROSPECTOR.

Function	Description
(CDR '⟨space⟩.⟨model⟩)	Returns property list of ⟨space⟩.⟨model⟩
(DO-SPACE '⟨space⟩.⟨model⟩)	Performs consultation from ⟨space⟩.⟨model⟩ and returns list of responses
(GET '⟨model⟩ 'topspace)	Returns topspace of ⟨model⟩
(GET-LIST '⟨space⟩.⟨model⟩ '⟨keyword⟩)	Returns value of ⟨keyword⟩ in ⟨space⟩.⟨model⟩
(GET-LIST-CHOICES '⟨space⟩.⟨model⟩ 'choices)	Returns list of choices for ⟨space⟩.⟨model⟩
(GET-SPACE '⟨space⟩.⟨model⟩ '⟨response⟩)	Returns space pointed by ⟨response⟩ for ⟨space⟩.⟨model⟩
(MUPROSPECTOR)	Runs muPROSPECTOR

THE muPROSPECTOR PROGRAM

The source listing of the current (1986) version of the muPROSPECTOR computer program is given in Appendix C. The program is written in the muLISP programming language, developed for use on microcomputers. A discussion of muLISP is beyond the scope of this paper; the interested reader should refer to the muLISP '85 Reference Manual (The Soft Warehouse, 1985) for details. LISP, from which muLISP derives, is the standard language of the artificial-intelligence research community. LISP and its derivatives continue to monopolize applications in this field.

USES AND LIMITATIONS

The muPROSPECTOR mineral consultant system is most useful at present for those evaluating the mineral potential of large areas in which the available data are of a reconnaissance nature. Such areas are likely to contain undiscovered deposits of a wide variety of deposit types. In these situations, the geologist could be at a loss to consider all of the possible deposit types that may be suggested by the data. There is the additional problem of expressing the degree of certainty associated with each evaluation. muPROSPECTOR provides the geologist with a set of rules that can be used to identify the types of deposits that may occur and the degree of certainty associated with each deposit type. The application of a fixed set of rules in evaluating the mineral potential of an area is more likely to produce results that are consistent from one area to the next, particularly when the evaluations are performed by more than one team of geologists. Furthermore, when differences between geologists occur, such differences can be reconciled more readily by isolating those factors giving rise to the differences. Quite often, a consensus can be reached within a short time.

A potential user should be aware of the limitations to muPROSPECTOR. The most obvious limitation is the incomplete knowledge base of deposit types. muPROSPECTOR cannot be expected to evaluate the potential for deposit types not represented in the knowledge base. Such a deficiency can be removed by incorporating these models at an early date. Moreover, existing models need continual review and updating, when appropriate. Potential users must also recognize that the certainty associated with a piece of evidence as it relates to a particular deposit model, or for that matter the certainty associated with different combinations of evidence, cannot yet be adequately represented. For now, the certainty is expressed verbally by using such terms as probable, possible, speculative, and so forth, thereby avoiding quantitative statements associated with formal probabilistic methods. Clearly, a more adequate representation of degree of certainty is a goal for muPROSPECTOR.

REFERENCES CITED

Duda, R.O., 1980, The PROSPECTOR system for mineral exploration: Menlo Park, Calif., Artificial Intelligence Cen-

ter, SRI International, Final Report, SRI Project 8172, 120 p.

___1981, Operation manual for the PROSPECTOR consultant system: Menlo Park, Calif., Artificial Intelligence Center, SRI International, Final Report, SRI Project 8172, 75 p.

Duda, R.O., Hart, P.E., Barrett, P., Gaschnig, J.G., Konlige, K., Reboh, R., and Slocum, J., 1978, Development of the PROSPECTOR consultant system for mineral exploration: Menlo Park, Calif., Artificial Intelligence Center, SRI International, Final Report, SRI Projects 5821 and 6415, 193 p.

McCammon, R.B., Boudette, E.L., Cameron, C.C., Cox, L.J., and Moench, R.H., 1984, An expert system for mineral resource assessment in the Sherbrooke-Lewiston 1 × 2 degree quadrangles, Maine, New Hampshire and Vermont: U.S. Geological Survey Open-File Report 84–751, 37 p.

Reboh, R., 1981, Knowledge engineering techniques and tools in the PROSPECTOR environment: Menlo Park, Calif., Artificial Intelligence Center, SRI International, Final Report, SRI Projects 5821, 6415, and 8172, 149 p.

The Soft Warehouse, 1985, muLISP-85 reference manual: Honolulu, Hawaii, 221 p.

APPENDIXES A–C

APPENDIX A

Session with muPROSPECTOR

The actual computer transcript of a session with muPROSPECTOR, given below, is designed to illustrate many of the features in its current state of implementation. In the transcript of the session, the responses to the questions appear on the line immediately below each question. The numbers enclosed within parentheses at the right side of the transcript were inserted after the session was terminated and are keyed to the numbered comments following the transcript.

Computer Transcript

```
A>muprospector nerap                                                    (1)

muPROSPECTOR --- Version II                                             (2)
U.S. Geological Survey
1-Nov-85     13:39:28
Do you want a transcript of this session (Y/N/?)                        (3)
Y

                * * NEW ENGLAND MINERAL RESOURCE APPRAISAL PROGRAM * *

This is a program to assist in the evaluation of environments favorable for the
occurrence of metallic deposits in New England. It is based on the occurrence
models used by Moench, Boudette, McCammon, Cameron, Cox, and others in their
assessment of the potential mineral resources in the Sherbrooke and Lewiston
two-degree sheets.
In using the program you will be asked to answer questions about the existence
of various kinds of evidence.  Answers to questions can be "Y ","N " or if you
don't know, "D ". To quit, enter "Q ".  In addition to supplying answers, you
can request information at any time with the following commands:

                ? -- Print a rephrased version of the question
                A -- Access database to assist in answering the question
                S -- Print a summary at this point in the consultation
                T -- Trace muPROSPECTOR's partial conclusions
                U -- Stop the trace
                W -- Show answer to a previous question
                B -- Break the consultation and put you in muLISP

 Press RETURN to continue ...
```

m u P R O S P E C T O R I I

TERRANE CLASSIFICATION

Number	terrane
1	Unknown
2	Metavolcanic
3	Metasedimentary
4	Ordovician-to-Carboniferous-plutonic
5	Jurassic-volcanic-plutonic
6	Surficial-material

Enter number or Q to quit:
4 (4)
The following is intended to aid in evaluating the Potential Mineral Resources in Ordovician to Carboniferous plutonic terranes in New England
1 -- For which of the following do you have any information?

1) Age of intrusive is Ordovician to Carboniferous (500 -290 m. y.)
2) Age of intrusive is Ordovician to Silurian (500 -410 m. y.)
3) Age of intrusive is Devonian (410 -360 m. y.)
4) Age of intrusive is Carboniferous (360 -290 m. y.)

Please enter one or more of the preceding numbers
(separated by blanks and terminating with a <CR>
2 3 (5)

The following questions relate to the potential resources of the intrusive whose age is Ordovician to Silurian

2 -- Can the rock be classified essentially as being of quartz-bearing intermediate composition, that is, granodiorite, tonalite or adamellite, and if visible, being composed of closely space phenocrysts in a microgranular aphanitic quartz-feldspar groundmass with a texture similar to fine aplite (Y/N/D/A/S/T/U/W/Q/B/?) :
Y (6)
3 -- Do you observe contacts with volcanic rocks, breccia fragments of the country rock, and/or numerous dikes and apophyses in the vicinity (Y/N/D/A/S/T/U/W/Q/B/?) :?
? (7)
Ore-related intrusions are always passive. That is, their contacts show stopping, intrusive brecciation, and veining. Also, the conditions listed in the question are evidence of a sub-volcanic environment

3 -- Do you observe contacts with volcanic rocks, breccia fragments of the country rock, and/or numerous dikes and apophyses in the vicinity (Y/N/D/A/S/T/U/W/Q/B/?) :

```
Y
 4 -- Do you observe abundant quartz-vein filled fractures, or more significant,
  abundant pyrite in closely space veinlets in the rock
 (Y/N/D/A/S/T/U/W/Q/B/?) :
T                                                                    (8)
Ok. please re - enter. 4 -- Do you observe abundant quartz-vein filled
 4 -- Do you observe abundant quartz-vein filled fractures, or more significant,
  abundant pyrite in closely spaced veinlets in the rock (Y/N/D/A/S/T/U/W/Q/B/?)
 (Y/N/D/A/S/T/U/W/Q/B/?) :
Y

 You have established:                                               (9)
      1) Age of intrusive is Ordovician to Silurian (500 -410 m. y.)
      2) Favorable porphyry type intrusive
      3) Favorable associated rock types
      4) Evidence of fracturing

So far, I have concluded that:

     5) Insufficient evidence at present indicates a Potential Resource for
Copper in a Porphyry Copper deposit

 5 -- Are there signs of alteration indicated by the presence of chlorite,
 epidote, actinolite, calcite, pyrite, or hematite (Y/N/D/A/S/T/U/W/Q/B/?) :
U                                                                    (10)
Trace off. please re - enter.
 5 -- Are there signs of alteration indicated by the presence of chlorite,
 epidote, actinolite, calcite, pyrite, or hematite (Y/N/D/A/S/T/U/W/Q/B/?) :
Y
 6 -- Is there evidence of clay replacement of plagioclase and are mafic
minerals or Al-rich minerals such as pyrophyllite or andalusite present
(Y/N/D/A/S/T/U/W/Q/B/?) :
D                                                                    (11)
 7 -- Do you have geochemical data from either stream sediments or pan
concentrates (Y/N/D/A/S/T/U/W/Q/B/?) :
Y
 8 -- Do you observe amounts of Cu, Pb, and Zn in excess of 100 ppm in stream
sediments and amounts of Ba in excess of 3000 ppm and Sr in excess of 1000 ppm
in heavy mineral concentrates at one location or a group of related locations
(Y/N/D/A/S/T/U/W/Q/B/?) :
?
 Although the geochemical expression of porphyry metal deposits is highly
variable, the concentrations of the above elements have been found extending out
to 10 km from known deposits
```

```
 8 -- Do you observe amounts of Cu, Pb, and Zn in excess of 100 ppm in stream
sediments and amounts of Ba in excess of 3000 ppm and Sr in excess of 1000 ppm
in heavy mineral concentrates at one location or a group of related locations
(Y/N/D/A/S/T/U/W/Q/B/?) :
?
 Although the geochemical expression of porphyry metal deposits is highly
variable, the concentrations of the above elements have been found extending out
to 10 km from known deposits

Y
 9 -- For which of the following do you have any information:
 1) No data
 2) Data on molybdenum in the rock
 3) Data on gold in the rock
 4) Data on copper and silver
 Please enter one or more of the preceding numbers
      (separated by blanks and terminating with a <CR>)
2 4

 10 -- Do the data suggest that molybdenum is present in amounts over 300 ppm
over a considerable volume of the intrusive rock (Y/N/D/A/S/T/U/W/Q/B/?) :
S                                                                    (12)

 You have established:
      1) Age of intrusive is Ordovician to Silurian (500 -410 m. y.)
      2) Favorable porphyry type intrusive
      3) Favorable associated rock types
      4) Evidence of fracturing
      5) Evidence of propylitic alteration
      6) Geochemical data available
      7) Presence of base metal anomalies
      8) Data on Molybdenum in the rock

You were uncertain about:

      9) Evidence of argillic alteration

So far, I have concluded that:

     10) Probable Potential Resource for Cu-Mo in Porphyry Copper Molybdenum-
         Rich Deposit
```

```
 10 -- Do the data suggest that molybdenum is present in amounts greater than
300 ppm over a considerable volume of the intrusive rock (Y/N/D/A/S/T/U/W/Q/B/?) :
N
 11 -- Do you have data that suggest that copper is present in amounts greater
than 3000 ppm and silver in amounts greater than 1 g/ t over a considerable
volume of the intrusive and country rock (Y/N/D/A/S/T/U/W/Q/B/?) :
W                                                                    (13)
Enter Question Number:
4                                                                    (14)
 4 -- Do you observe abundant quartz-vein filled fractures, or more significant,
abundant pyrite in closely space veinlets in the rock  ANS :  Y   Any more
questions?
N
please re - enter.
 11 -- Do you have data that suggest that copper is present in amounts greater
than 3000 ppm and silver in amounts greater than 1 g/ t over a considerable
volume of the intrusive and country rock (Y/N/D/A/S/T/U/W/Q/B/?) :
Y
 Y The following questions relate to the potential resources of the intrusive
whose age is Devonian                                                (15)
 12 -- Can the intrusive best be described as a pink to gray, variably
porphyritic, pegmatitic or two-mica granite (Y/N/D/A/S/T/U/W/Q/B/?) :
Y
 13 -- Do you observe major faults in the vicinity or is the intrusive highly
fractured (Y/N/D/A/S/T/U/W/Q/B/?) :
?
 Extensive faulting and fracturing of the source rock for uranium is considered
 favorable for localized uranium concentrations

 13 -- Do you observe major faults in the vicinity or is the intrusive
highly fractured (Y/N/D/A/S/T/U/W/Q/B/?) :
Y
 14 -- Can you identify potash-enriched zones within the intrusive
 (Y/N/D/A/S/T/U/W/Q/B/?) :
D
 --------------------
 Based on your responses, my evaluation is as follows:               (16)

 1) Probable Potential Resource for Cu-Mo in Porphyry Copper Molybdenum-Rich
    Deposit
 2) Definite Potential Resource for Copper in Porphyry Copper Deposit
 3) Speculative Potential Resource for Uranium in Vein Deposit
```

```
You have established:

    4) Age of intrusive is Ordovician to Silurian (500-410 m. y.)
    5) Favorable porphyry type intrusive
    6) Favorable associated rock types
    7) Evidence of fracturing
    8) Evidence of propylitic alteration
    9) Geochemical data available
   10) Presence of base metal anomalies
   11) Data on molybdenum in the rock
   12) No porphyry copper molybdenum-rich grade material
   13) Data on copper and silver
   14) Porphyry copper grade material
   15) Age of intrusive is Devonian (410-360 m.y.)
   16) Favorable uranium source rock
   17) Evidence of fractures

You were uncertain about:

   18) Evidence of argillic alteration
   19) Presence of potash-enriched zone
Do you wish to consider another terrane or another area (Y/N/Q/B/?):
N

1111339.TRN                                                          (17)
```

Comments on the Computer Transcript

(Comments are keyed to numbers inserted in the right-hand side of the computer transcript above.)

1. The user initiates muPROSPECTOR for the models in NERAP.

2. The program prints the version of muPROSPECTOR being executed and date and time that the program is being run.

3. The user wants a transcript of the session.

4. The user selects the Ordovician-to-Carboniferous-plutonic terrane model.

5. The user selects choices 2 and 3.

6. The user answers "Y" to the question.

7. The user enters the "?" command and receives more information.

8. The user enters the "T" command that turns on the trace.

9. After answering the question "Y", the user sees the partial conclusions reached at thus far in the consultation.

10. The user turns off the trace.

11. The user answers "D" (don't know) to the question.

12. The user enters the "S" command and receives a summary of the conclusions reached thus far in the consultation.

13. The user enters the "W" command to see the response to an earlier question.

14. The user wants to see the response to question number 4.

15. An announcement precedes question number 12.

16. All questions relating to the pieces of evidence that muPROSPECTOR needs to reach at a conclusion have been asked, and an evaluation is made.

17. The transcript of the session is stored in the file named 1111339.TRN; the prefix of the file name refers to the month-day-hour-minutes when session ended; in this example, November 1st at 1:39 p.m.

Appendix B

BACKUS-NAUR FORM (BNF) DESCRIPTION OF THE FORMAL LANGUAGE FOR EXTERNAL REPRESENTATION OF muPROSPECTOR MODELS

This appendix contains a modified Backus-Naur Form description of the syntax of muPROSPECTOR's knowledge representation language, which is used for representing models in external files. The syntax described here is the one accepted by muPROSPECTOR's parser.

The following conventions apply:

- Terminal symbols are written in uppercase.
- Nonterminal symbols are either written in uppercase and enclosed in angled brackets or implicitly defined by a phrase written within double quotes.
- Alternate choices are enclosed in parentheses.
- Optional items are enclosed in square brackets.
- Items repeated at least once are enclosed in curly brackets.
- Unless stated otherwise, terminals must be separated by a <delimiter> (defined below).

<model definition> ::= MODEL <model name>
   TOPSPACE <space name> [<space definition>] <stop>
<model name> ::= <character string>
<character string> ::= "a sequence of characters except a <delimiter character>"
<delimiter character> ::= ("space" "tab" "carriage return" "line feed"
   "comma" "period")

```
<space name> ::= <character string>

<text> ::= /* <text body> */

<text body> ::= "a sequence of arbitrary characters, except the
   sequence '*/'"

<space definition> ::= space <space name> <text section> [(<optional
   section>)]

<text section> ::= [<text entry>]

<text entry> ::= (<description entry> [<announce entry>] [<explanation
    entry>])

<description entry> ::= DESC ( <space name>  <text> )

<announce entry> ::= ANNOUNCE ( <space name>  <text> )

<explanation entry> ::= EXPL  ( <space name>  <text> )

<inference entry> ::= INF  ( <space name> <text>  )

<choices entry> ::= CHOICES  [ <space name> ]

<noun-phrase entry> ::= NOUN-PHRASE  ( <space name>  <text> )

<question entry> ::= QUES  ( <space name>  <text> )

<yes entry> ::= YES  <space name>

<no entry> ::= NO <space name>

<default entry> ::= ( DEF  <space name>  <no entry> )
```

APPENDIX C

Program listing of muPROSPECTOR

The following is the source listing of muPROSPECTOR. muPROSPECTOR is written in the muLISP programming language. For information about muLISP, the reader is referred to the muLISP reference manual (The Soft Warehouse, 1985).

```
; File: MUPROSPE.LSP          11/01/85          U.S. Geological Survey

            The muPROSPECTOR Energy and Mineral Resources Consultant

(LOOP (PRIN1 '*) (EVAL (READ)) ((NULL RDS)))

(SETQ *PRINT-DOWNCASE*)

(SETQ *PRINT-ESCAPE*)

(SETQ *READ-UPCASE*)

(SETQ *IGNORE-CASE*)

(SETQ *DRIBBLE*)

(SETQ READ-CHAR T)

(SET-BREAK-CHARS (DELETE '- (GET-BREAK-CHARS) 'EQUAL))

(SET-BREAK-CHARS (DELETE '_ (GET-BREAK-CHARS) 'EQUAL))

(SETQ CHCLST '(choices))

(SETQ HDRLST '(HELLO PURPOSE TITLE FILENAMES OBJLST BANNER OBJECT ATTRIBUTE))

(SETQ MONTHS '((1 . Jan)
  (2 . Feb)
  (3 . Mar)
  (4 . Apr)
  (5 . May)
  (6 . Jun)
  (7 . Jul)
  (8 . Aug)
  (9 . Sept)
  (10 . Oct)
  (11 . Nov)
  (12 . Dec) ))
```

```
(SETQ PARLIS '("("))

(SETQ PCTLIS '("," ";" ":" ")" "." "'" "/" "?" "!"))

(SETQ PLIST '(desc ques expl inf noun-phrase announce database ))

(SETQ POSNEG '((Evidence . "Lack of evidence")
  (Adequate . Inadequate)
  (Presence . Absence)
  (Absence . Presence)
  (Favorable . Unfavorable)
  (Indication . "No indication")
  (Identified . Unidentified)
  (Have . "Have not")
  (Differentiated . Undifferentiated)
  (Estimable . Inestimable)
  (Familiar . Unfamiliar) ))

(SETQ PTRLST '(yes no def))

(SETQ QUOTLIS '("            "") )

(SETQ RESPONSE '((Y . yes)
  (N . no)
  (D . def) ))

(SETQ TERMINATOR '(?))

(DEFUN CENTER (MSG)
  (SPACES (TRUNCATE (- (LINELENGTH) (LENGTH MSG)) 2))
  (PRINC MSG) (TERPRI))

(DEFUN CONTINUE ( ; Local:
    RDS)
  (PRTSENT '(Press RETURN to continue "..."))
  (FLUSH) )

(DEFUN DO-CHOICES (CHOICES ; Local:
    LST N CHCS)
  (SETQ CHCS CHOICES)
  (PUSH CHOICES LST)
  (SETQ N 0)
  (LOOP
    ((NULL CHOICES)
      (LIST (REVERSE LST)) )
    (SETQ N (ADD1 N))
    (PUSH (CONS SPACE (CONS 'C (CONS N CHCS))) *RESULTS*)
    (PUSH (DO-SPACE (NTH (- (POP CHOICES) 1) (GET-LIST-CHOICES SPACE
        'choices))) LST) ) )
```

```
(DEFUN DO-SPACE (SPACE)
  ((EQ (CAR (FLAGP SPACE 'inf)) 'inf)
    (PUSH (CONS SPACE (CONS 'I 'T)) *RESULTS*)
    NIL )
  ((GET SPACE 'choices)
    (DO-CHOICES (GET-CHOICES SPACE)) )
  (SETQ ANSWR (GET-YESNO SPACE))
  (CONS ANSWR (DO-SPACE (GET-SPACE SPACE ANSWR))) )

(DEFUN DRIBBLE (CHAR)
  ((NOT (NULL *DRIBBLE*))
      (SETQ ECHO NIL)
      (PRINT CHAR)
      (SETQ ECHO T) ))

(DEFUN EQLIST (LST1 LST2)
  ((NULL LST1)
    (NULL LST2) )
  ((NULL LST2) NIL)
  ((NOT (EQ (CAR LST1) (CAR LST2))) NIL)
  (EQLIST (CDR LST1) (CDR LST2)) )

(DEFUN EVALUATE (RESULT SP ; Local:
    STR ANS EV EVSP EST UNC N)
  ((NULL RESULT)
    (PRTSENT '(No results yet) 2) )
  (SETQ STR RESULT)
  (LOOP
    (SETQ ANS (POP STR))
    ((NULL ANS)
      (SETQ EV (REMOVE-DUPLISTS (REVERSE EV))))
    ( ((AND
          (EQ (CADR ANS) 'I)
          (NULL (MEMBER (CAR ANS) EVSP 'EQUAL)) )
        (PUSH (GET-LIST (CAR ANS) 'inf) EV)
        (PUSH (CAR ANS) EVSP) ) ) )
  (SETQ N 0)
  ( ((NULL SP)
      (SEPARATOR)
      (PRTSENT (APPEND '(Based on your responses) '("," my
          evaluation is as follows:)) 2)
      (SETQ EV (REVERSE EV))
      (LOOP
        ((NULL EV))
        (SETQ N (ADD1 N))
        (PRTSENT (APPEND (LIST (PACK* N '")" )) (POP EV)) 2) )) )
```

```
  (LOOP
    (SETQ ANS (POP RESULT))
    ((NULL ANS)
      (TERPRI)
      (SETQ EST (REMOVE-DUPLISTS EST))
      (PRTSENT '(You have established:) 2)
      (LOOP
        ((NULL EST))
        (SPACES 5)
        (SETQ N (ADD1 N))
        (PRTSENT (APPEND (LIST (PACK* N '")" )) (POP EST)) 1) )
      ((NOT (NULL UNC))
        (TERPRI)
        (SETQ UNC (REMOVE-DUPLISTS UNC))
        (PRTSENT '(You were uncertain about:) 2)
        (LOOP
          ((NULL UNC))
          (SPACES 5)
          (SETQ N (ADD1 N))
          (PRTSENT (APPEND (LIST (PACK* N '")" )) (POP UNC)) 1) ) ) )
    (LOOP
      ((AND (EQ (CADR ANS) 'Q) (EQ (CDDR ANS) 'D))
        (PUSH (GET-LIST (CAR ANS) 'desc) UNC) )
      ((AND (EQ (CADR ANS) 'Q) (EQ (CDDR ANS) 'Y))
        (PUSH (GET-LIST (CAR ANS) 'desc) EST) )
      ((EQ (CADR ANS) 'C)
        (PUSH (GET-LIST (NTH (- (NTH (- (CADDR ANS) 1) (CDDDR ANS)) 1)
          (GET-LIST-CHOICES (CAR ANS) 'choices)) 'noun-phrase) EST) )
      ((AND (EQ (CADR ANS) 'Q) (EQ (CDDR ANS) 'N))
        (PUSH (NO-ANSWER (GET-LIST (CAR ANS) 'desc)) EST) )
      ((EQ (CADR ANS) 'I)) ) )
  ((NOT (NULL SP))
    (SETQ *EVSP* EVSP)
    (GET-DEPTH SP)
    (SETQ EVSP (REVERSE (PRUNE-LIST (REVERSE *EVSP*))))
    (TERPRI)
    (PRTSENT '("So far," I have concluded that:) 2)
    (LOOP
      ((NULL EVSP)
        (TERPRI)
        (TERPRI) )
      (SPACES 5)
      (SETQ N (ADD1 N))
      (PRTSENT (APPEND (LIST (PACK* N '")" )) (GET-LIST (POP EVSP) 'inf)) 1)
      ) ) )

(DEFUN EVAL-CHOICES (CHOICES)
  (LOOP
    ((NULL CHOICES))
    (GET-DEPTH (POP CHOICES)) ) )
```

```
(DEFUN FIND-LENGTH (MSG)
  (SETQ N 0)
  (LOOP
    ((NULL MSG) N)
    (SETQ N (+ N (LENGTH (POP MSG)))) ) )

(DEFUN FLUSH NIL
  (LOOP
    ((EQ (READ-CHAR) CR) )))

(DEFUN GET-CHOICE (LST ; Local:
    CHAR RDS READ)
  (LOOP
    (SETQ CHAR (RATOM))
    (TERPRI)
    (FLUSH)
    (DRIBBLE CHAR)
    ((OR (EQ CHAR 'Q) (EQ CHAR 'q))
      (SYSTEM) )
    (SETQ NAM (CADR (ASSOC CHAR FILENAMES 'EQUAL)))
    ((AND
        (MEMBER CHAR LST 'EQUAL)
        (NOT (EQ NAM 'NONE)) ) NAM)
    ( ((EQ NAM 'NONE)
        (PRTSENT '(This part of the knowledge base not yet completed ".") 2)
            ) )
    (PRTSENT '(please re - enter ".") 1) ) )

(DEFUN GET-CHOICES (SPACE ; Local:
    LST N)
  ( ((NOT (EQ (GET SPACE 'announce) NIL))
      (PRTSENT (GET-LIST SPACE 'announce) 2) ) )
  (PRTSENT (APPEND (LIST (+ QUESTN 1) '--)
    '(For which of the following do you have any information :)) 2)
  (SETQ LST (GET-LIST-CHOICES SPACE 'choices))
  (SETQ QUESTN (ADD1 QUESTN))
  (SETQ N 0)
  (LOOP
    ((NULL LST))
    (SETQ N (+ N 1))
    (PRTSENT (APPEND (LIST N) '(")") (GET-LIST (POP LST) 'noun-phrase)) 1) )
  (TERPRI)
```

```
  (LOOP
    (SETQ LST NIL)
    (PRTSENT '(Please enter one or more of the preceding numbers) 1)
    (SPACES 5)
    (PRTSENT '("(separated" by blanks and terminating with a <CR>))
    (SET-BREAK-CHARS (LIST CR) T)
    (LOOP
      (SETQ CHAR (RATOM))
      (DRIBBLE CHAR)
      ( ((OR (EQ CHAR 'B) (EQ CHAR 'b))
          (BREAK SPACE "Current Space") ) )
      ((OR (EQ CHAR 'Q) (EQ CHAR 'q))
        (SYSTEM) )
      ((EQ CHAR CR)
        (TERPRI) )
      ( ((MEMBER CHAR LST 'EQUAL))
        (PUSH CHAR LST) ) )
      (SET-BREAK-CHARS (DELETE CR (GET-BREAK-CHARS)))
      ((TEST-LST LST N)
      (REVERSE LST) )
    (PRTSENT '(please re - enter ".") 1) ) )

(DEFUN GET-DEFAULT (SP KEY ; Local:
    LST ECHO)
  (SETQ LST (GET-LIST SP KEY LST))
  ( ((NOT (EQ LST NIL) LST)
      (SETQ LST 'N) ) ) )

(DEFUN GET-DEPTH (SP)
  (LOOP
    ((EQ (CAR (FLAGP SP 'inf)) 'inf)
      (SETQ *EVSP* (APPEND (LIST SP) *EVSP*)) )
    ((GET SP 'choices)
      (EVAL-CHOICES (GET-LIST-CHOICES SP 'choices)) )
    (SETQ SP (GET-SPACE SP 'no)) ) )

(DEFUN GET-FILENAMES (FILNAM LST)
  (LOOP
    ((NULL FILNAM)
      (REVERSE LST) )
    (PUSH (CAR (POP FILNAM)) LST) ) )

(DEFUN GET-LIST (SP KEY ; Local :
    LST FIRST ECHO)
  (SETQ POINT (GET SP KEY))
  ((EQ POINT NIL)
    (SETQ LST '(No further explanation available at this time)) )
```

```
  (RDS (PACK* (CAR POINT) '.MDL))
  (READPTR (CDR POINT))
  (RATOM)
  (RATOM)
  (LOOP
    (SETQ NEXT (RATOM))
    ((AND
        (EQ NEXT '*)
        (EQ (READ-CHAR T) '/) )
      (PROG1
        (REVERSE LST)
        (RDS) ) )
    (SETQ LST (CONS NEXT LST)) ) )

(DEFUN GET-LIST-CHOICES (SP KEY ; Local:
    LST FIRST ECHO)
  (SETQ POINT (GET SP KEY))
  (RDS (PACK* (CAR POINT) '.MDL))
  (READPTR (CDR POINT))
  (RATOM)
  (RATOM)
  (LOOP
    (SETQ NEXT (RATOM))
    ( ((EQ (PEEK-CHAR T) '.)
        (RATOM)
        (SETQ NEXT (PACK* NEXT '. (RATOM))) )
      ( ((NOT (EQ NEXT '*))
          (SETQ NEXT (PACK* NEXT '. (CAR POINT))) )) )
    ((AND
        (EQ NEXT '*)
        (EQ (READ-CHAR T) '/) )
      (PROG1
        (REVERSE LST)
        (RDS) ) )
    (SETQ LST (CONS NEXT LST)) ) )

(DEFUN GET-NUMBER (M ; Local :
    READ RDS)
  (LOOP
    (PRTMSG '(Enter Question Number:))
    (SETQ M (RATOM))
    (TERPRI)
    (FLUSH)
    (DRIBBLE M)
    ((AND (INTEGERP M) (> M 0) (<= M QUESTN)) M)
    (PRTSENT '(out of range - please re-enter ".") 1) ) )
```

```
(DEFUN GET-OBJECT (; Local:
    LST1 LST2 NUM)
  (RDS)
  (TERPRI)
  (CLEAR-SCREEN)
  (CENTER "m u P R O S P E C T O R   I I")
  (TERPRI)
  (SPACES (TRUNCATE (- (LINELENGTH) (FIND-LENGTH BANNER)) 2))
  (PRTSENT BANNER 2)
  (SPACES (TRUNCATE (- (LINELENGTH) 46) 2))
  (PRTSENT (APPEND '("Number                ") OBJECT) 1)
  (TERPRI)
  (SETQ LST1 OBJLST)
  (SETQ LST2)
  (SETQ NUM 1)
  (LOOP
    ((EQ LST1))
    (SPACES (TRUNCATE (- (LINELENGTH) 43) 2))
    (PUSH (PRINC NUM) LST2)
    (SETQ NUM (+ NUM 1))
    (SPACES 1)
    ( (((< (LINELENGTH) 50))
      (SPACES 3) )
    (PRINT (POP LST1)) )
  (TERPRI 2)
  (PRTMSG '(Enter number or Q to quit:))
  (GET-CHOICE (GET-FILENAMES FILENAMES)) )

(DEFUN GET-QUESTION (M ; Local:
    N LST)
  (SETQ LST (REVERSE *RESULTS*))
  (SETQ N 0)
  (LOOP
    ( ((OR (AND (EQ (CADAR LST) 'C)
        (EQ (CADDAR LST) 1) )
        (EQ (CADAR LST) 'Q))
          (SETQ N (ADD1 N)) ))
    ((EQ N M) (POP LST))
    (POP LST) ))
```

```
(DEFUN GET-RESPONSE (SPACE LST HLP ; Local:
    CHAR READ RDS)
  (LOOP
    (PRTSENT LST)
    (SETQ CHAR (RATOM))
    (TERPRI)
    (FLUSH)
    (DRIBBLE CHAR)
    ( ((OR (EQ CHAR 'T) (EQ CHAR 't))
        (SETQ *TRACE* T)
        (PRTMSG '("Ok.")) ) )
    ( ((OR (EQ CHAR 'U) (EQ CHAR 'u))
        (SETQ *TRACE*)
        (PRTMSG '("Trace off.")) ) )
    ( ((OR (EQ CHAR 'B) (EQ CHAR 'b))
        (BREAK SPACE "Current Space") ) )
    ( ((OR (EQ CHAR 'W) (EQ CHAR 'w))
        (SHOW-QUESTION) ) )
    ((OR (EQ CHAR 'Q) (EQ CHAR 'q))
      (SYSTEM) )
    ((OR (EQ CHAR 'Y) (EQ CHAR 'y))
      'Y )
    ((OR (EQ CHAR 'N) (EQ CHAR 'n))
      'N )
    ((OR (EQ CHAR 'D) (EQ CHAR 'd))
      'D )
    ((OR (EQ CHAR 'S) (EQ CHAR 's))
      'S )
    ((OR (EQ CHAR 'A) (EQ CHAR 'a))
      (SETQ TMP (GET-LIST SPACE 'database))
      ((EQ (CAR TMP) 'No)
        (PRTSENT '(Database to assist in answering this question not
            available) 2)
        (GET-RESPONSE SPACE LST HLP CHAR READ READ-CHAR RDS TMP) )
      (EXECUTE "COMMAND.COM" (PACK (CONS "/C " TMP)))
      (SETQ READ-CHAR 'T)
      (GET-RESPONSE SPACE LST HLP CHAR READ READ-CHAR RDS TMP) )
    ( ((EQ CHAR '?)
        (PRTSENT HLP 2) )
      (PRTSENT '(please re - enter ".") 1) ) ) )

(DEFUN GET-SPACE (SP ANS ; Local:
    READPTR RDS ECHO TMP)
  (SETQ KEY (CADR (ASSOC ANS RESPONSE 'EQUAL)))
  (SETQ POINT (GET SP KEY))
  ( ((EQ POINT NIL)
      (SETQ POINT (GET SP 'no)) ))
  (RDS (PACK* (CAR POINT) '.MDL))
```

```
  (READPTR (CDR POINT))
  (SETQ TMP (RATOM))
  (SETQ *RDS* (CAR POINT))
  ( ((EQ (PEEK-CHAR T) '.)
     (RATOM)
     (SETQ *RDS* (RATOM))) )
  (PACK* TMP '. *RDS*) )

(DEFUN GET-YESNO (SPACE ANS)
  ((GET (GET SPACE 'PQUES) 'ASKED)
    (PUSH (CONS SPACE (CONS 'Q (GET (GET SPACE 'PQUES) 'ANS)))
        *RESULTS*)
    (GET (GET SPACE 'PQUES) 'ANS) )
  ( ((NOT (EQ (GET SPACE 'announce) NIL))
     (PRTSENT (GET-LIST SPACE 'announce) 2) ) )
  (LOOP
    (SETQ ANS (GET-RESPONSE SPACE (APPEND (LIST (+ QUESTN 1) '--)
        (GET-LIST SPACE 'ques) '("(Y/N/D/A/S/T/U/W/Q/B/?) :"))
           (GET-LIST SPACE 'expl)))
    ((NOT (EQ ANS 'S))
      (PUSH (CONS SPACE (CONS 'Q ANS)) *RESULTS*)
      (PUT (GET SPACE 'PQUES) 'ASKED 'T)
      (PUT (GET SPACE 'PQUES) 'ANS ANS)
      (SETQ QUESTN (+ QUESTN 1))
      ( ((EQ *TRACE* 'T)
         (EVALUATE *RESULTS* SPACE) ))
      ANS )
    (CLEAR-SCREEN)
    (EVALUATE *RESULTS* SPACE) ) )

(DEFUN HEADER NIL
  (RDS)
  (REGISTER 0 10752)
  (INTERRUPT 33)
  (SETQ YEAR (- (REGISTER 2) 1900))
  (SETQ MONTH-DAY (DIVIDE (REGISTER 3) 256))
  (REGISTER 0 11264)
  (INTERRUPT 33)
  (SETQ HOUR-MINUTES (DIVIDE (REGISTER 2) 256))
  (SETQ SECONDS (DIVIDE (REGISTER 3) 256))
  (SETQ MINUTES (CDR (DIVIDE (REGISTER 2) 256)))
  (SETQ SECONDS (CAR SECONDS))
  ((< MINUTES 10)
    (SETQ MINUTES (PACK (LIST '0 MINUTES))) )
  ((< SECONDS 10)
    (SETQ SECONDS (PACK (LIST '0 SECONDS))) )
  (SETQ DATE-LINE (LIST (CDR MONTH-DAY) '- (CDR (ASSOC (CAR MONTH-DAY)
      MONTHS 'EQUAL)) '- YEAR "    " (CAR HOUR-MINUTES) ': MINUTES ': SECONDS))
  (PRINT "muPROSPECTOR  ---  Version II")
  (PRINT "U.S. Geological Survey")
  (PRINT (PACK DATE-LINE)) )
```

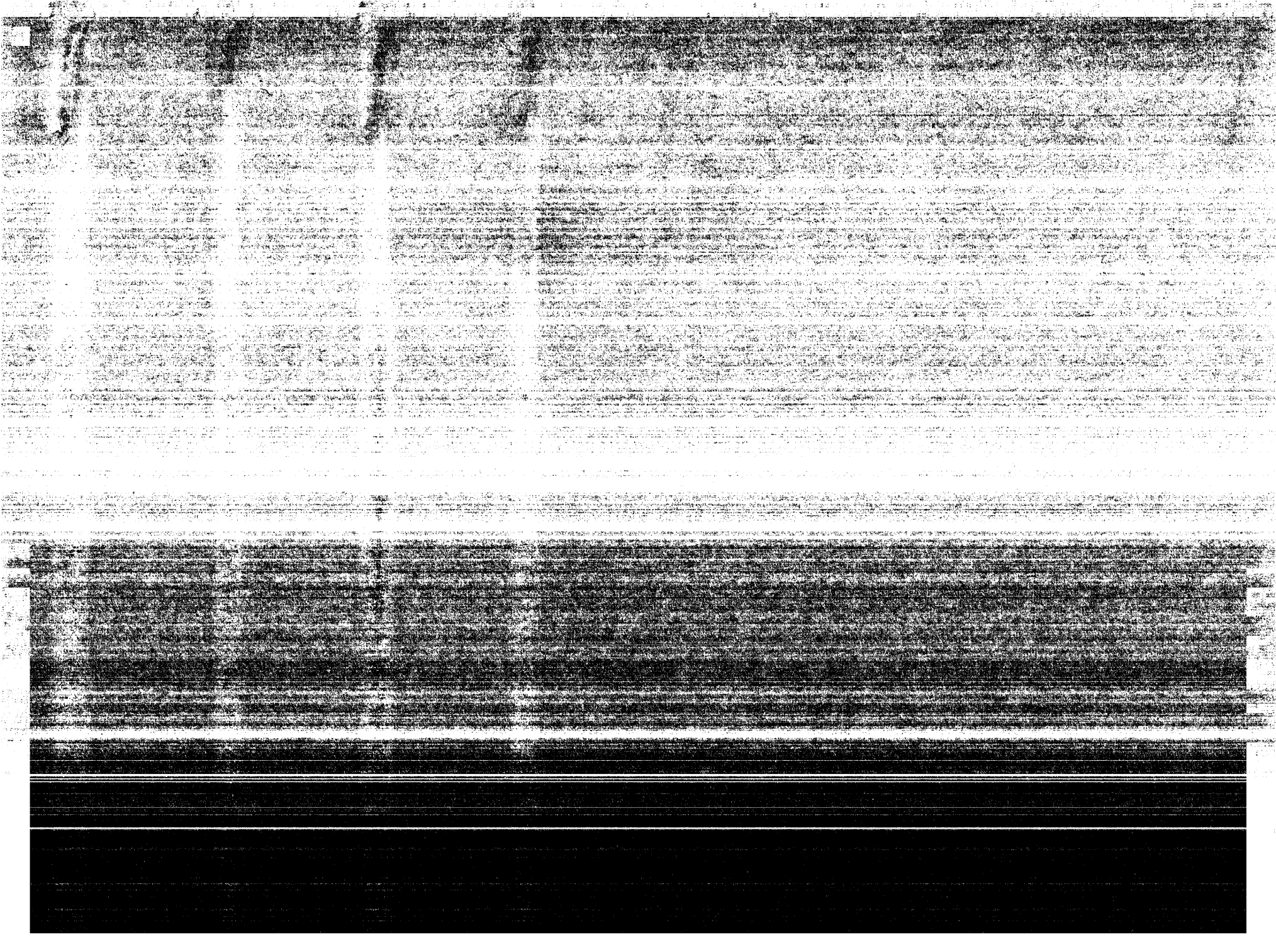
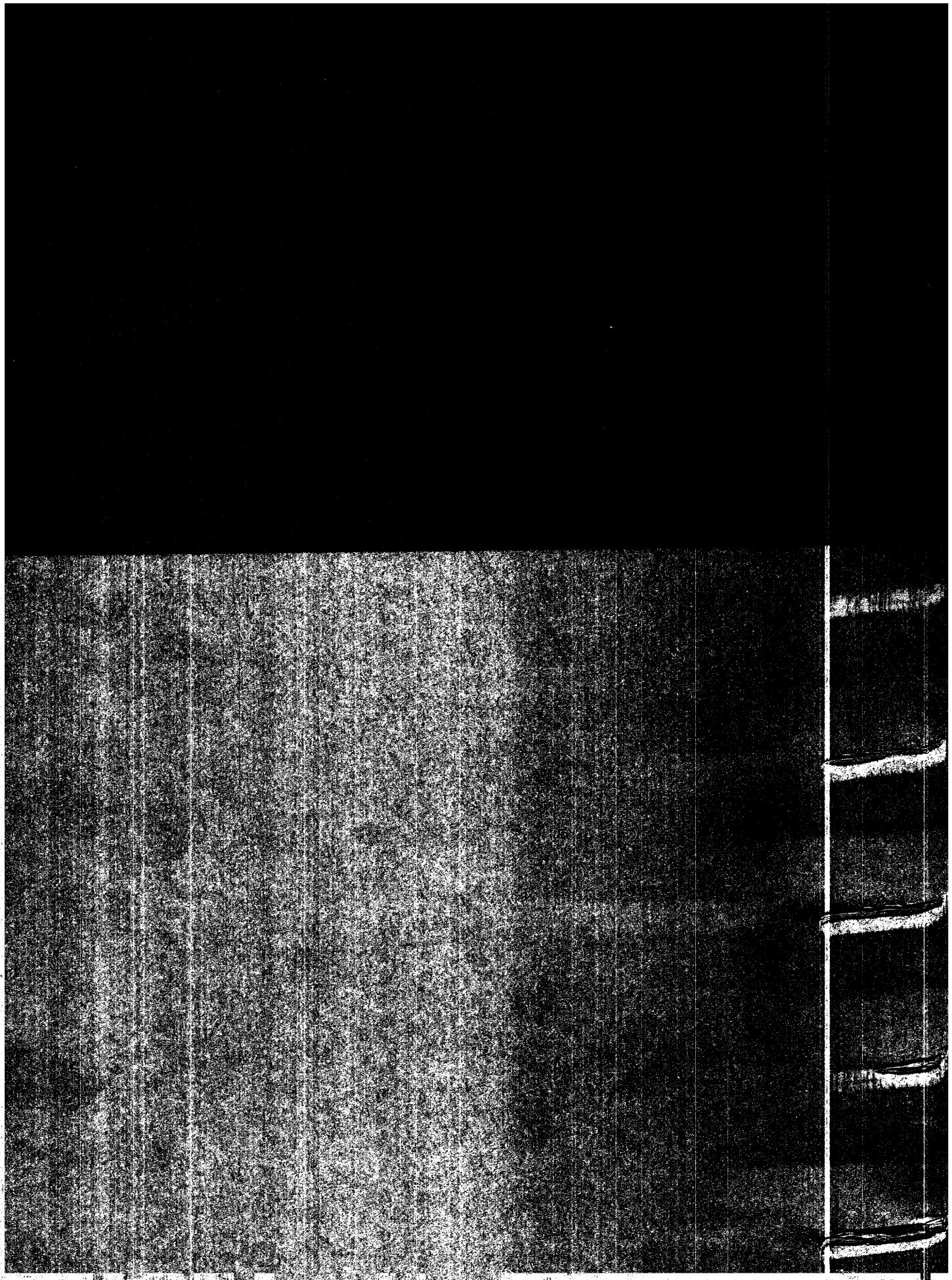
```
(DEFUN INSTRUCTIONS (RDS)
  (PRTSENT '(In using the program you will be asked to answer questions
      about the existence of various kinds of evidence "." Answers to
      questions can be       "Y
 " ","  "N " "," or if you dont know ","
           "D                " "." To quit "," enter
 "Q " "." In addition to supplying
      answers "," you can request information at any time with the
      following commands:) 2)
  (SPACES 10)
  (PRTSENT '("? -- Print" a rephrased version of the question) 1)
  (SPACES 10)
  (PRTSENT '("A -- Access" database to assist in answering the question) 1)
  (SPACES 10)
  (PRTSENT '("S -- Print" a summary at this point in the consultation) 1)
  (SPACES 10)
  (PRTSENT '("T -- Let's" you see "muPROSPECTOR's" partial conclusions) 1)
  (SPACES 10)
  (PRTSENT '("U -- Stops" the trace) 1)
  (SPACES 10)
  (PRTSENT '("W -- Show" answer to a previous question) 1)
  (SPACES 10)
  (PRTSENT '("B -- Break" the consultation and put you in MuLISP) 2)
  (CONTINUE) )

(DEFUN LOAD-MODEL (FILNAM ; Local:
    ECHO RDS)
  (RDS (PACK* FILNAM '.MDL))
  (LOOP
    (SETQ EXPN (RATOM))
    ((EQ EXPN 'STOP)
      (RDS) )
    ( ((EQ EXPN 'model)
        (PUSH (RATOM) *LOADED-MODELS*)
        (PUT FILNAM 'model (CAR *LOADED-MODELS*)) ) )
    ( ((EQ EXPN 'topspace)
        (PUT (CAR *LOADED-MODELS*) 'topspace (PACK* (RATOM) '. FILNAM)) ) )
    ( ((EQ EXPN 'space)
        (SETQ SPACE (PACK* (RATOM) '. FILNAM))
        (RPLACD SPACE NIL)
        (PUT SPACE 'PQUES SPACE)
        (PUSH SPACE *MODEL-SPACES*) ) )
    ( ((MEMBER EXPN PLIST 'EQUAL)
        (SETQ PROP EXPN)
        ( ((EQ PROP 'inf)
            (FLAG SPACE PROP) ) )
        (SETQ POINT (CONS FILNAM (READPTR)) )
        (SETQ EXPN (RATOM))
```

```
        (SETQ PFLG (MEMBER (PACK* EXPN '. FILNAM) *MODEL-SPACES* 'EQUAL))
        ( ((NOT (EQ PFLG NIL))
            (SETQ POINT (GET (CAR PFLG) PROP))
            ( ((EQ PROP 'ques)
                (PUT SPACE 'PQUES (CAR PFLG)) ) ) ) )
        (PUT SPACE PROP POINT)
        ( ((AND
              (EQ EXPN '/)
              (EQ (RATOM) '*) )
            (PASS-EXPN) ) ) )
    ( ((MEMBER EXPN PTRLST 'EQUAL)
        (PUT SPACE EXPN (CONS FILNAM (READPTR)))
        (RATOM)
        ( ((EQ (PEEK-CHAR T) '.)
            (RATOM)
            (SETQ EXPN (RATOM))
            ( ((NOT (OR (MEMBER EXPN *ADD-MODELS* 'EQUAL)
                        (MEMBER EXPN *LOADED-MODELS* 'EQUAL)))
                (PUSH EXPN *ADD-MODELS*) ) ))) ))
    ( ((MEMBER EXPN CHCLST 'EQUAL)
        (PUT SPACE EXPN (CONS FILNAM (READPTR)))
        (RATOM)
        (RATOM)
        (LOOP
          ((EQ (RATOM) '*)
            (RATOM) )
          ( ((EQ (PEEK-CHAR T) '.)
              (RATOM)
              (SETQ EXPN (RATOM))
              ( ((NOT (OR (MEMBER EXPN *ADD-MODELS* 'EQUAL)
                          (MEMBER EXPN *LOADED-MODELS* 'EQUAL)))
                  (PUSH EXPN *ADD-MODELS*))) )) ) )) )
  ((NOT (NULL *ADD-MODELS*))
    (LOAD-MODEL (POP *ADD-MODELS*)) ) )

(DEFUN MUPROSPECTOR NIL
  (RDS)
  (CLEAR-SCREEN)
  (HEADER)
  (WAIT 1000)
  (SETQ CR (ASCII 13))
  (READ-HDRFILE)
  (SETQ FILENAMES (SEQ-FILENAMES FILENAMES))
  (SETQ *DRIBBLE* (QUERY '(Do you want a transcript of this session
    "(Y/N/?)" )
    '(This will create a file that you can review at your leisure)))
  ( ((EQ *DRIBBLE* 'T)
      (SETQ ECHO T)
      (WRS (SETQ *DRIBBLE* (PACK* (CAR MONTH-DAY) (CDR MONTH-DAY)
         (CAR HOUR-MINUTES)
         (CDR HOUR-MINUTES) '.TRN)) ) ))
```

```
  (CLEAR-SCREEN)
  (TERPRI 2)
  (SPACES (TRUNCATE (- (LINELENGTH) (FIND-LENGTH TITLE) 6) 2))
  (PRTSENT TITLE 2)
  (SETQ *FILNAM* '*FILNAM*)
  (SETQ *ADD-MODELS*)
  (SETQ *LOADED-MODELS*)
  (SETQ *MODEL-SPACES*)
  (PRTSENT PURPOSE 1)
  (INSTRUCTIONS)
  (LOOP
    ((LOOP
        (SETQ *FILNAM* (GET-OBJECT))
        ((RDS (PACK* *FILNAM* '.MDL))
          (RDS) )
        (PRTSENT '(File NOT found) 2) ))
    ( ((NOT (MEMBER *FILNAM* *LOADED-MODELS*))
        (LOAD-MODEL *FILNAM*)
        (SETQ *MODEL-SPACES* (REVERSE *MODEL-SPACES*)) ) )
    (CLEAR-SCREEN)
    (SETQ QUESTN 0)
    (SETQ *RESULTS*)
    (SETQ *RDS* *FILNAM*)
    (UNDO *MODEL-SPACES*)
    (SETQ *TRACE*)
    (DO-SPACE (GET (GET *FILNAM* 'model) 'topspace))
    (CLEAR-SCREEN)
    (EVALUATE *RESULTS*)
    (RDS)
    (TERPRI)
    (TERPRI)
    ((NOT (QUERY (APPEND '(Do you wish to consider another) OBJECT '(or
        another) ATTRIBUTE '("(Y/N/Q/B/?):")) '(You may wish to
        reconsider your answers)))) )
  ( ((NOT (NULL *DRIBBLE*))
      (TERPRI)
      (PRTSENT (LIST *DRIBBLE*) 1) ))
  (WRS)
  (SYSTEM) )

(DEFUN NO-ANSWER (LST ; Local:
    FIX)
  (SETQ FIX (CADR (ASSOC (CAR LST) POSNEG 'EQUAL)))
  ((NOT (EQ FIX NIL))
    (CONS FIX (CDR LST)) )
  (CONS 'No (CONS (PACK (CONS (STRING-DOWNCASE (CAR (UNPACK (CAR LST))))
    (CDR (UNPACK (CAR LST))))) (CDR LST)  ) ) )

(DEFUN PAGE-FULL NIL
  ((<= (ROW) 21) NIL)
  (TERPRI)
  (CLEAR-SCREEN) )
```

```
(DEFUN PASS-EXPN NIL
  (LOOP
    ((AND
        (EQ (RATOM) '*)
        (EQ (READ-CHAR T) '/) )) ) )

(DEFUN PRETERPRI (WORD)
  ((> (+ (SPACES) (LENGTH WORD)) (LINELENGTH))
    (TERPRI) ) )

(DEFUN PRTMSG (LST)
  (LOOP
    (PRINC (POP LST))
    ((NULL LST))
    ( ((< (SPACES) (LINELENGTH))
        (SPACES 1) ) ) ) )

(DEFUN PRTSENT (LST NUM ; Local:
    WORD)
  (LOOP
    ((NULL LST))
    (SETQ WORD (POP LST))
    ( ((MEMBER WORD QUOTLIS)
      (LOOP
        ((MEMBER (CAR LST) QUOTLIS)
          (SETQ WORD (PACK* WORD (POP LST))) )
        (SETQ WORD (PACK* WORD (POP LST))) ) ) )
    ( ((MEMBER WORD PARLIS)
      (SETQ WORD (PACK* WORD (POP LST))) ) )
    ( ((OR (NULL LST) (MEMBER (CAR LST) PCTLIS) )
        (PRETERPRI WORD) ) )
    ( ((MEMBER WORD PCTLIS))
      (SPACES 1) )
    (PRINC WORD) )
  ((PLUSP NUM)
    (TERPRI NUM) )
  ((EQ (ADD1 (SPACES)) (LINELENGTH))
    (TERPRI) )
  (SPACES 2) )

(DEFUN PRUNE-LIST (LST)
  (SETQ LST1 NIL)
  (LOOP
    ((NULL LST) LST1)
    ( ((MEMBER (CAR LST) (CDR LST) 'EQUAL) NIL)
      (PUSH (CAR LST) LST1) )
    (SETQ LST (CDR LST)) ) )
```

```
(DEFUN READ-HDRFILE NIL
  (SETQ FILENAMES '("No FILENAMES"))
  (SETQ HELLO '("You have no HELLO"))
  (SETQ PURPOSE '("You have no PURPOSE"))
  (SETQ TITLE '("You have no TITLE"))
  (SETQ OBJLST '("You have no OBJLST"))
  (SETQ BANNER '("You have no BANNER"))
  (SETQ OBJECT '("You have no OBJECT"))
  (SETQ ATTRIBUTE '("You have no ATTRIBUTE"))
  ((RDS 'MUPROSPECTOR.HDR)
    (LOOP
      (SETQ EXPN (RATOM))
      ((EQ EXPN 'STOP)
        (RDS) )
      ( ((MEMBER EXPN HDRLST 'EQUAL)
          (SET (CAR (MEMBER EXPN HDRLST 'EQUAL)) (READ-LIST)) ) ) ) )
  (PRTSENT '(File NOT found) 2)
  (SYSTEM) )

(DEFUN READ-LIST (NEXT LST)
  (RATOM)
  (RATOM)
  (LOOP
    (SETQ NEXT (RATOM))
    ((AND
        (EQ NEXT '*)
        (EQ (READ-CHAR T) '/) )
      (REVERSE LST) )
    (SETQ LST (CONS NEXT LST)) ) )

(DEFUN REMOVE (LST1 LST2)
  ((NULL LST2) NIL)
  ((EQLIST LST1 (CAR LST2))
    (REMOVE LST1 (CDR LST2)) )
  (CONS (CAR LST2) (REMOVE LST1 (CDR LST2))) )

(DEFUN REMOVE-DUPLISTS (LST LST1)
  ((NULL (CADR LST)) LST)
  (LOOP
    ((NULL LST)
      (REVERSE LST1) )
    (SETQ LST1 (CONS (CAR LST) LST1))
    (SETQ LST (REMOVE (CAR LST) (CDR LST))) ) )

(DEFUN REV (X)
  (COND
    ((NULL X) NIL)
    (T (APPEND (REV (CDR X)) (REV2 (CAR X)))) ) )
```

```
(DEFUN REV2 (Y)
  (COND
    ((ATOM Y)
      (LIST Y) )
    ((ATOM (CAR Y))
      (LIST (REV Y)) )
    (T (LIST (APPEND (REV (CDR Y)) (REV2 (CAR Y))))) ) )

(DEFUN QUERY (LST HLP ; Local:
    CHAR READ RDS)
  (LOOP
    (PRTSENT LST)
    (SETQ CHAR (RATOM))
    (TERPRI)
    (FLUSH)
    (DRIBBLE CHAR)
    ( ((OR (EQ CHAR 'B) (EQ CHAR 'b))
        (BREAK *RDS* "Current Model") ) )
    ((OR (EQ CHAR 'Q) (EQ CHAR 'q))
      (SYSTEM) )
    ((OR (EQ CHAR 'Y) (EQ CHAR 'y)))
    ((OR (EQ CHAR 'N) (EQ CHAR 'n)) NIL)
    ( ((EQ CHAR '?)
        (PRTSENT HLP 2) )
      (PRTSENT '(please re - enter ".") 1) ) ) )

(DEFUN SEPARATOR NIL
  (TERPRI)
  (PRTSENT '(--------------------) 1) )

(DEFUN SEQ-FILENAMES (LST ; Local:
    LST1 N)
  (SETQ N 0)
  (LOOP
    ((NULL LST)
      (REVERSE LST1) )
    (PUSH (CONS (ADD1 N) (POP LST)) LST1)
    (SETQ N (ADD1 N)) ) )
```

```
(DEFUN SHOW-QUESTION (M ; Local:
    ELEM SP LST N)
  (LOOP
    (SETQ M (GET-NUMBER))
    (SETQ ELEM (GET-QUESTION M))
    ( ((EQ (CADR ELEM) 'Q)
       (PRTSENT (APPEND (LIST M '--) (GET-LIST (CAR ELEM) 'ques)
           '(" ANS : ") (LIST (CDDR ELEM)) )) ))
    ( ((EQ (CADR ELEM) 'C)
       (PRTSENT (APPEND (LIST M '--)
          '(For which of the following do you have any information :)) 2)
       (SETQ LST (GET-LIST-CHOICES (CAR ELEM) 'choices))
       (SETQ N 0)
       (LOOP
         ((NULL LST))
         (SETQ N (+ N 1))
         (PRTSENT (APPEND (LIST N) '(")")
             (GET-LIST (POP LST) 'noun-phrase)) 1) )
       (TERPRI)
       (PRTSENT (APPEND '(" CHOICES : ") (CDDDR ELEM))) ))
    ((NOT (QUERY '(Any more ?)))) ) )

(DEFUN TEST-LST (LST N ; Local:
    OBJ)
  ((NULL LST) NIL)
  (LOOP
    ((NULL LST) T)
    (SETQ OBJ (POP LST))
    ((OR
       (NOT (PLUSP OBJ))
       (> OBJ N) ) NIL) ) )

(DEFUN UNDO (LST)
  (LOOP
    ((NULL LST))
    (PUT (POP LST) 'ASKED NIL) ) )

(DEFUN WAIT (LEN)
  (LOOP
    ((EQ LEN 1))
    (SETQ LEN (- LEN 1)) ) )
(RDS)
```