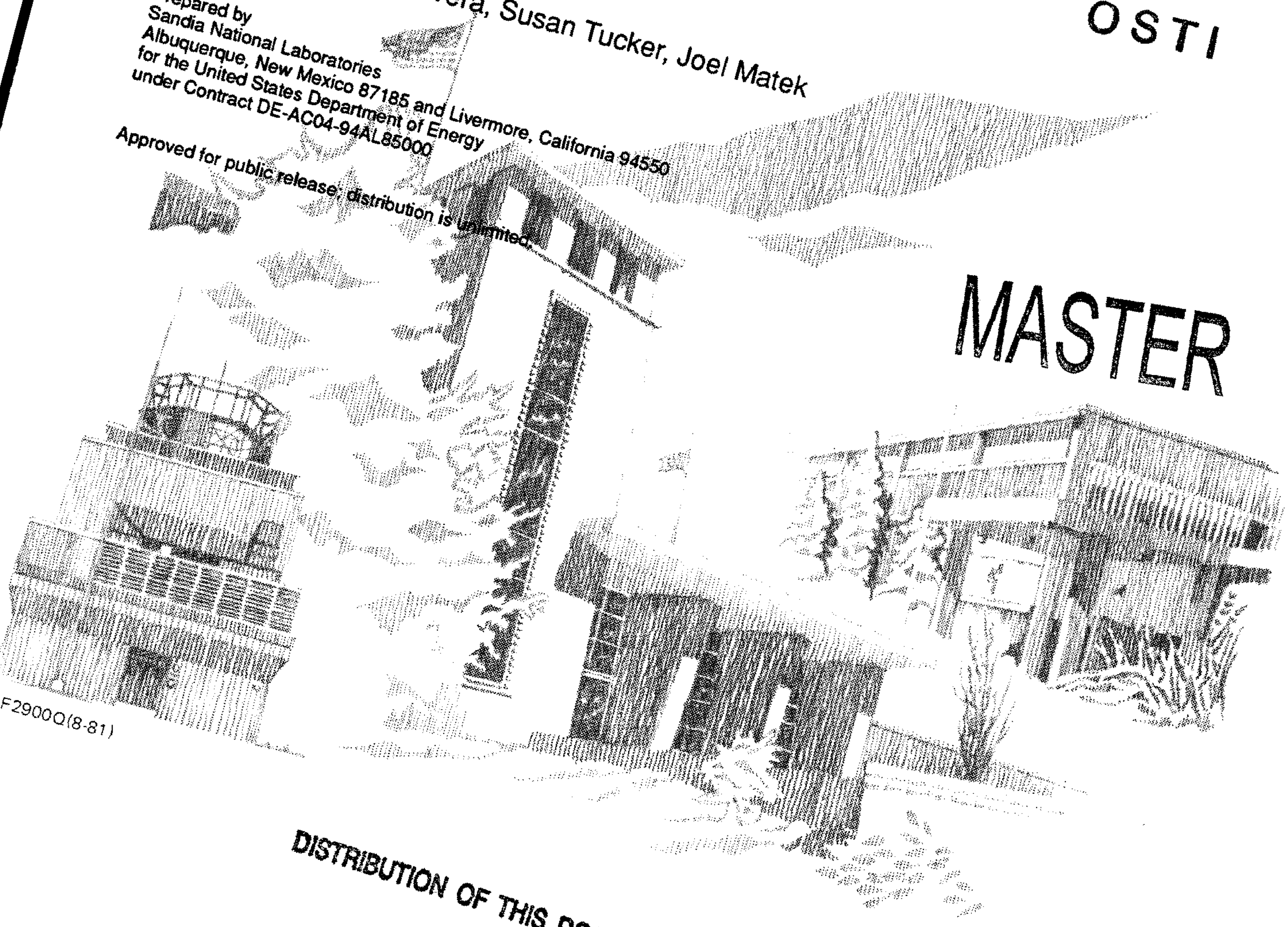
SANDIA REPORT
SAND96–1995 • UC–810
Unlimited Release
Printed August 1996

Knowledge Assistant for Robotic Environmental Characterization

John Feddema, Jill Rivera, Susan Tucker, Joel Matek

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND96-1995
Unlimited Release
Printed August 1996

Distribution
UC-810

Knowledge Assistant for Robotic Environmental Characterization

John Feddema
Intelligent Systems, Sensors & Controls Department

Jill Rivera
Advanced Engineering & Manufacturing Software Development Department

Susan Tucker
Manufacturing Control Subsystems Department

Joel Matek
SAIC

Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT

A prototype sensor fusion framework called the "Knowledge Assistant" has been developed and tested on a gantry robot at Sandia National Laboratories. This Knowledge Assistant guides the robot operator during the planning, execution, and post analysis stages of the characterization process. During the planning stage, the Knowledge Assistant suggests robot paths and speeds based on knowledge of sensors available and their physical characteristics. During execution, the Knowledge Assistant coordinates the collection of data through a data acquisition "specialist." During execution and postanalysis, the Knowledge Assistant sends raw data to other "specialists," which include statistical pattern recognition software, a neural network, and model-based search software. After the specialists return their results, the Knowledge Assistant consolidates the information and returns a report to the robot control system where the sensed objects and their attributes (e.g., estimated dimensions, weight, material composition, etc.) are displayed in the world model. This report highlights the major components of this system.

Acknowledgements

The authors owe a special thanks to Charles Little and Chris Wilson for providing the rapid world modelling system, visualization tools, and the MAPER interface; Prof. Ed Hensel and his students Jennifer Macy and Brandon Donahue for their help with MAPER; and Dave Fugelso for providing valuable communications software between modules.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Contents

Figures

Tables

1.0 INTRODUCTION

Over the past five years, Sandia National Laboratories (SNL) has been developing robotically deployed site characterization systems for DOE facilities such as Hanford and Idaho National Engineering Laboratory [1,2]. While these systems are good at gathering and storing large volumes of sensor data, final analysis of the data still requires a team of chemists and physicists. The objective of the work presented in this report is to develop a framework and a set of general tools that will simplify and speed up sensor data analysis and warn the operator of impending dangers or malfunctions. The software and algorithms developed should reduce the time and expense of the characterization process by minimizing the number of on-site technical personnel required to collect and interpret large volumes of complex, multivariate sensor data. This generic sensor fusion framework could be applied across a wide range of characterization activities including buried waste, underground storage tanks, laboratory analysis, and decommissioned facilities.

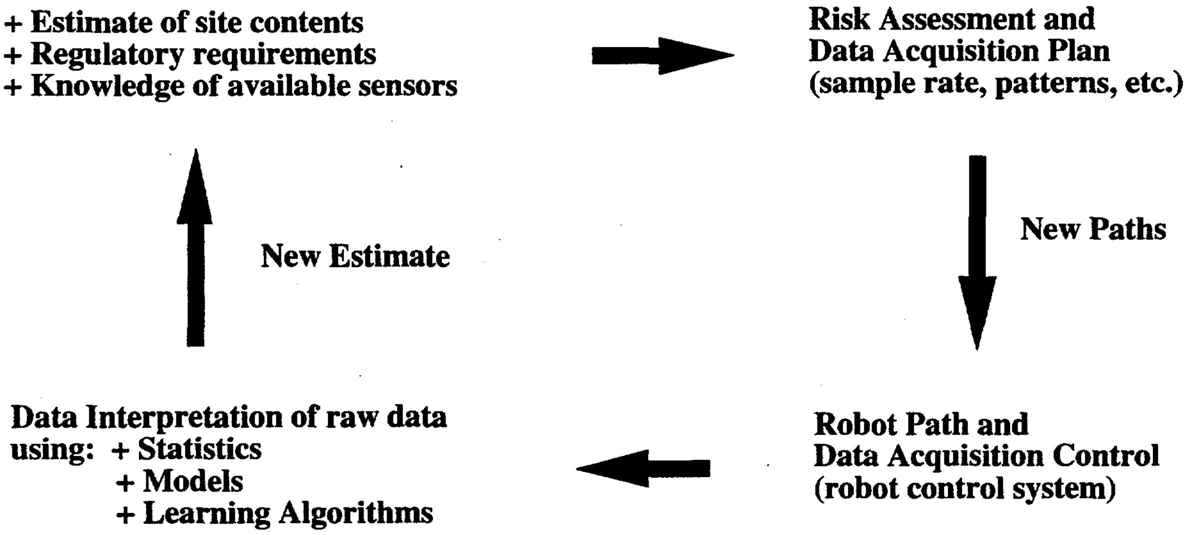


Figure 1. Robotic site characterization life cycle.

A sensor fusion (or data interpretation) framework could play a vital role in automating the characterization process. Figure 1 illustrates the robotic site characterization life cycle. The data interpretation module starts with knowledge of regulatory requirements, knowledge of the available sensors, and an estimate of the site contents. From this information, an expert system formulates a risk assessment and data acquisition plan. These plans are reviewed and modified by the site manager. The data acquisition plan includes sampling rates of the equipment and spacing and patterns of scans, which are directly translated into robot paths and instrumentation control sequences. While the robot is performing the motion and the data acquisition system is collecting the data, the data interpretation module looks at the data using various statistical, model-based, and learning tools. A new estimate of the site contents is continuously generated and rules on regulatory requirements are used to modify the risk assessment and data acquisition plan. As risk increases, the operator is warned; and if risk reaches a certain threshold, the operation is automatically stopped.

When investigating how a sensor fusion framework might be structured, we first analyzed the various stages of a data interpretation process. After discussing this issue with several experts in the field, the following general stages of data interpretation became apparent.

1. Low level data manipulation. For example, these include interpolators, data croppers, filtering algorithms, and transformations.
2. Extraction of features from the data either using signal processing techniques or physical sensor models. For example, the peaks and areas of spectral data, such as Gas Chromatography (GC), Raman, infrared, and gamma spectroscopy, are used to characterize data. Another example is the extraction of features within an aerial image to locate landmarks on a buried waste site.
3. Classification of data using techniques such as Bayesian hypothesis testing, Dempster-Shafer, Fuzzy Logic, and Neural Networks.
4. Heuristic expert system rules to guide the previous levels, make high-level control decisions, provide operator guidance, and provide early warnings and diagnostics.

Two pictorial examples of the information flow within a data interpretation framework are shown in Figures 2 and 3. Many of the early stages are sensor specific. However, if formulated correctly, many of the higher-level stages are reusable across many different sensing modalities. For example, many of the spectral analysis tools are common for GC, gamma, or Raman data interpretation. Also, the same classification tools can be used for classifying GC spectral as can be used for identifying barrels from GPR (Ground Penetrating Radar) data.

Based on this analysis, we began development of a "Knowledge Assistant" sensor fusion framework, which contains an expert system at the top level and various specialists at the lower levels (Figure 4). The Knowledge Assistant helps the operator of a robotic characterization system, much like a co-pilot helps a pilot navigate an airplane. The Knowledge Assistant performs the job of the chemist and physicist by determining how to scan over the environment based on the

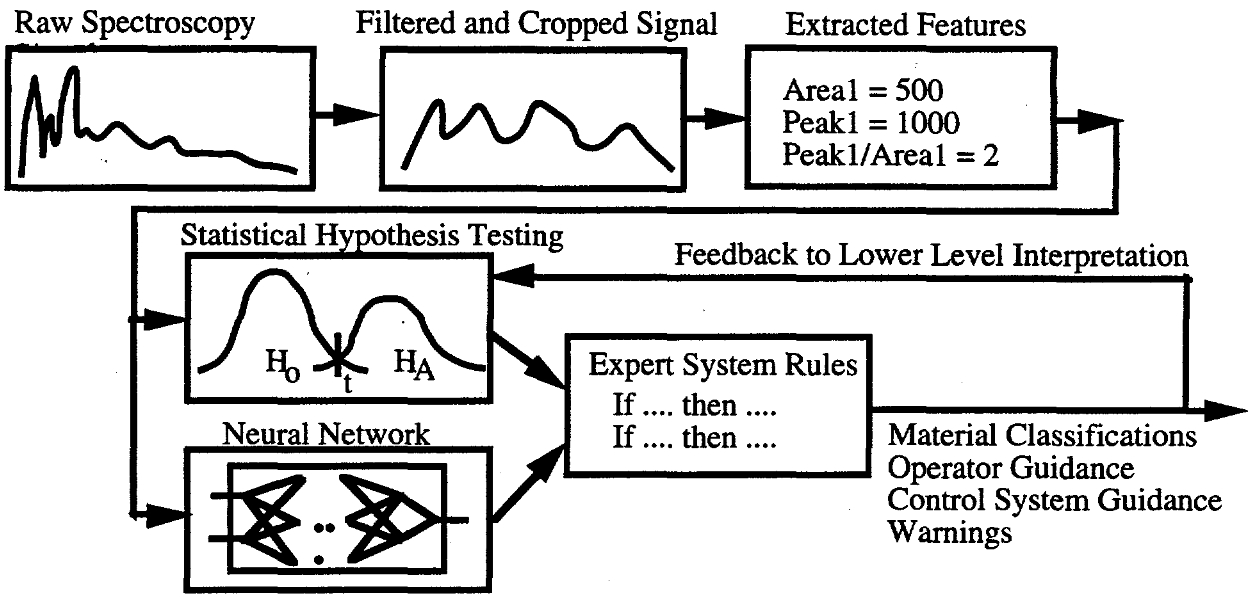


Figure 2. Data interpretation with statistical, learning, and expert system reasoning.

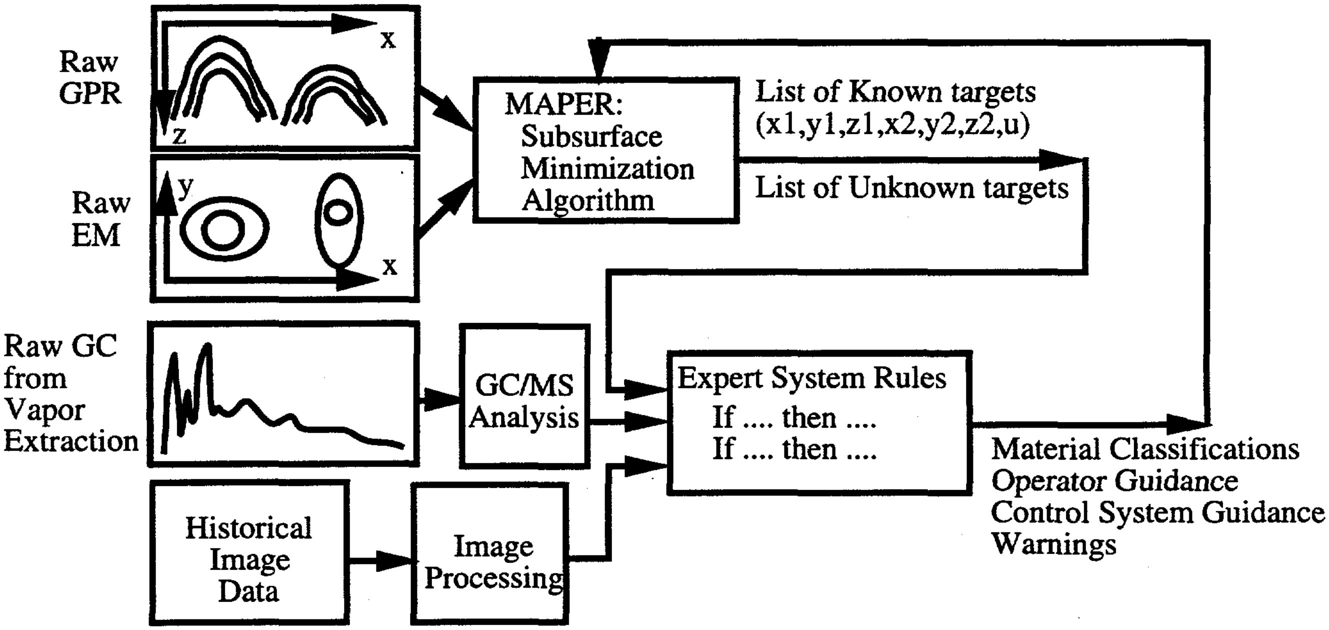


Figure 3. Data interpretation of subsurface mapping data.

sensor platform available and the regulatory requirements. The Knowledge Assistant also assists in the collection and postanalysis of the data.

Aiding the Knowledge Assistant are various specialists, which can be reconfigured based on the application. When making decisions, the Knowledge Assistant directs information to the lower levels and provides the final fusion of the results from the specialists. As an initial test, we developed four specialists:

- Data-Acquisition Specialist
- Statistical Pattern-Recognition Specialist
- Neural-Network Classifier Specialist
- Gradient-Based Model Fitting Specialist

This last specialist is New Mexico State University's Multi-sensor Analysis Program for Environmental Restoration (MAPER) system, and more detailed information can be found in References 3 and 4.

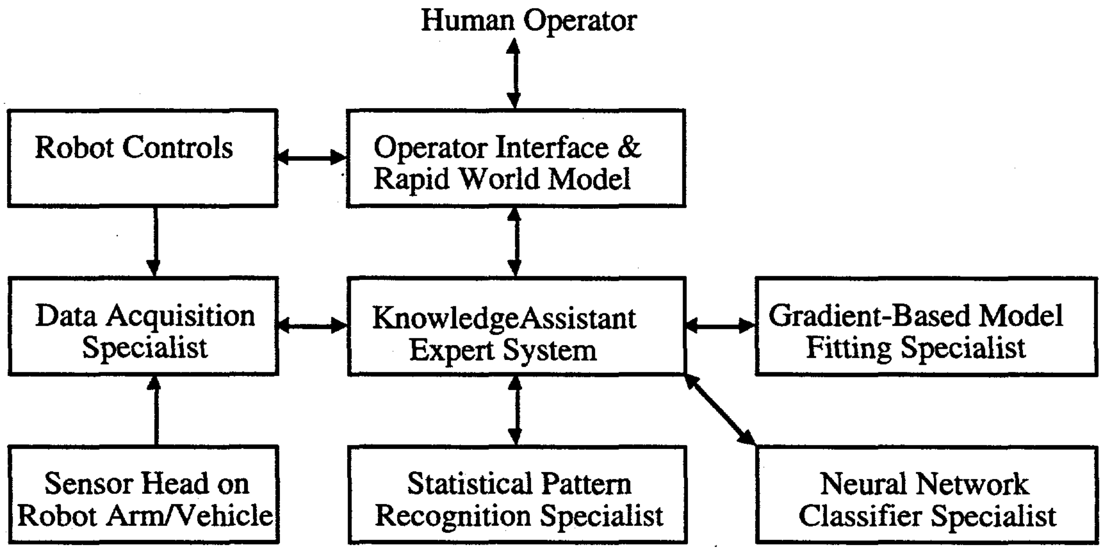

Human Operator
Robot Controls
Operator Interface & Rapid World Model
Data Acquisition Specialist
KnowledgeAssistant Expert System
Gradient-Based Model Fitting Specialist
Sensor Head on Robot Arm/Vehicle
Statistical Pattern Recognition Specialist
Neural Network Classifier Specialist

Figure 4. Knowledge Assistant architecture.

The next three sections discuss the Knowledge Assistant, the Statistical Pattern Recognition Specialist, and the Neural Network Classifier Specialist. As yet another example of how this technology can be applied in the environmental arena, Section 5.0 describes how the expert system was used to assess gas chromatography data. Conclusions and suggestions for future work are described in the final section.

2.0 KNOWLEDGE ASSISTANT

The Knowledge Assistant was developed on a real-time expert system. Through the expert system, heuristic reasoning in the form of "if-then" rules can be added to the system's knowledge base. For example, natural language "if-then" rules within the expert system have been used to determine the cause of gas chromatography errors (e.g., sample too concentrated, column pressure too low, etc.) from symptoms in the data (e.g., shifts in retention time, rising baseline, etc.). The Knowledge Assistant also directs the flow of data to the appropriate analysis routines. For example, when accessing gas chromatography data, the expert system sends the raw spectral data to signal processing routines, which look for symptoms. When analyzing magnetometer data, the expert system sends the data to a gradient search algorithm, which fits the raw sensor data to a model. Expert system rules are also used to set the initial boundary conditions and residual thresholds of the search. When categorizing soils, the expert system directs data to a statistical pattern recognition routine, which characterizes soils based on tip force, pore pressure, viscosity, and pH sensor readings from a cone penetrometer.

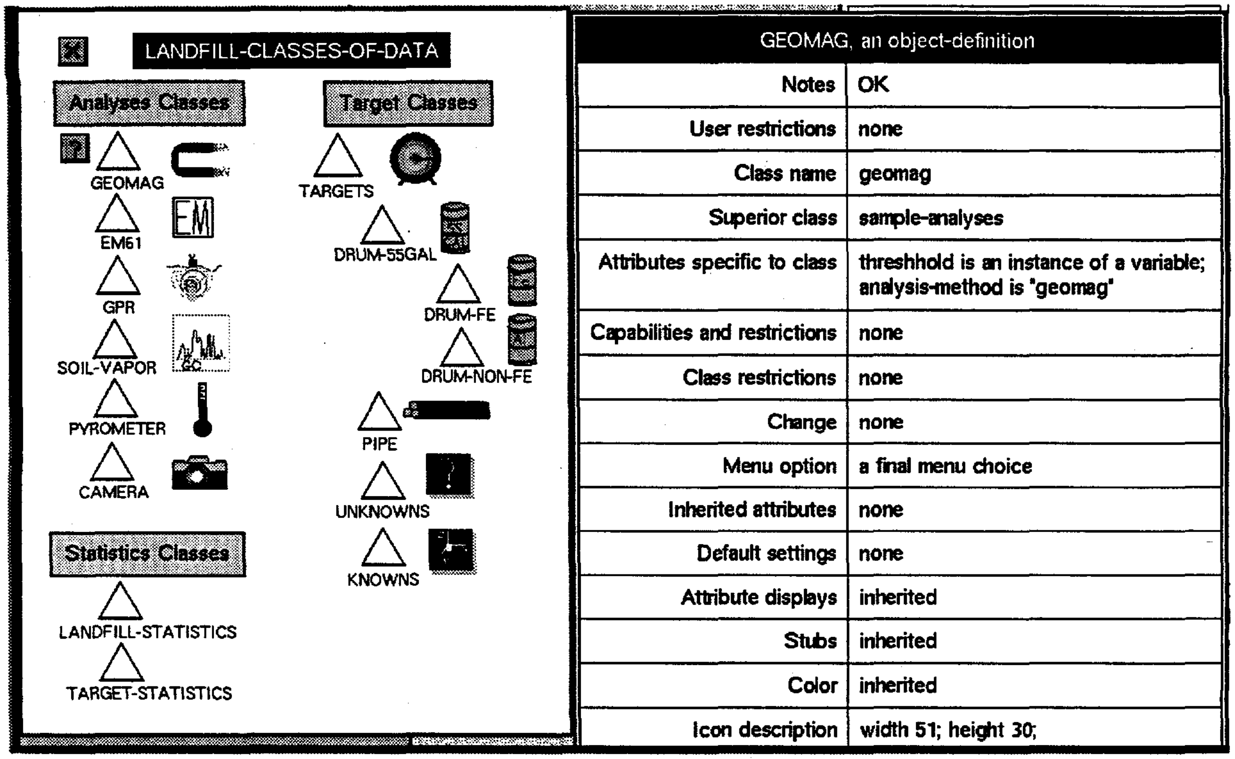
The Knowledge Assistant was built using Gensym's G2 real-time expert system shell [5]. The shell provides:

- an object-oriented approach to representing components in a workspace
- a graphical user interface
- a variety of data collection schemes
- a natural language interface for rules and procedures
- forward and backward chaining of rules.

The implementation of an object-oriented methodology creates a model of the application consisting of classes and subclasses with inherited attributes (variables or constants) and behaviors (functions). Objects with the same data attributes and behaviors are grouped into a class. A class is a broadly defined group of objects and a subclass is a specifically defined group of a class. Each subclass inherits all of the properties of its superclass and adds its own unique properties. The development of classes and subclasses greatly reduces repetition within the software and is one of the main advantages of an object-oriented system. The Knowledge Assistant has numerous classes and subclasses such as:

class: sample-analysis	subclasses: geomagnetic, pyrometer
class: target	subclasses: drum-55gal, pipe, known, unknown
class: sensor-targets	subclasses: geomagnetic, pyrometer.

Each object of a class is referred to as an instance. An instance is created either from user input, data collection, or a rule. Examples of classes and objects pertaining to landfills and an example attribute table are shown in Figure 5.



GEOMAG, an object-definition	
Notes	OK
User restrictions	none
Class name	geomag
Superior class	sample-analyses
Attributes specific to class	threshhold is an instance of a variable; analysis-method is "geomag"
Capabilities and restrictions	none
Class restrictions	none
Change	none
Menu option	a final menu choice
Inherited attributes	none
Default settings	none
Attribute displays	inherited
Stubs	inherited
Color	inherited
Icon description	width 51; height 30;

Figure 5. Example landfill objects with attribute table.

The graphical user interface (GUI) provides the means for user input. The Knowledge Assistant queries the user for sensor selection information and scanning information such as resolution and scan heights. The queries are in the form of radio boxes (Figure 6). This increases the reliability of the user input. Tables of attribute values can be displayed with simple mouse clicks. Help screens are available through the GUI. Also, the GUI provides the user the ability to abort the process at anytime. The Knowledge Assistant consists of several modules, e.g., a statistical module, a landfill module. The GUI can display windows, queries, charts, graphs, and reports from various modules at one time and provide continuity between the modules.

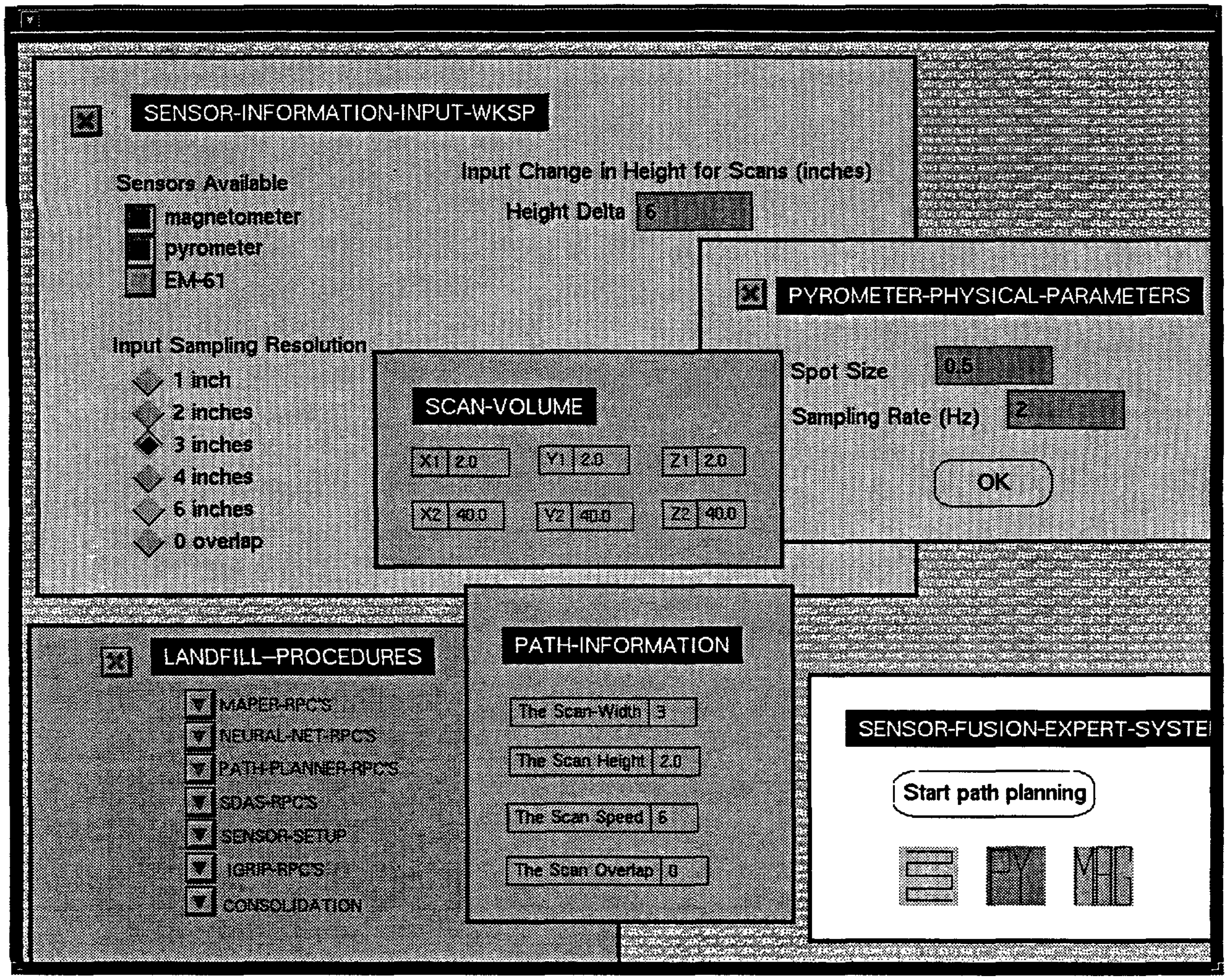
The modularity of the Knowledge Assistant allows for ease of reapplication and modification of the code. Instance information is passed quickly between modules. Modules can be added effortlessly.

The data-collection schemes within G2 include time-based, event-based, and unsolicited. The Knowledge Assistant relies on unsolicited data collection. Unsolicited data includes the sensor scanning region from the world model; data file names from the data acquisition specialist, MAPER, and the neural nets; and statistical information from MAPER. The data is processed by rules as it is received. The rules are written in a natural language, which allows for ease of understanding and modifying the rules. The rules currently in the Knowledge Assistant are forward chaining rules. Forward chaining infers that as an attribute receives a value or a series of attributes receive values, an outcome is determined.

To test the capabilities of the system, we presented the system with the task of robotically scanning over a mock buried waste site with a sensor package containing a magnetometer, pyrometer, and camera. The objective was to map out the environment, display it on a graphical screen, and list attributes (such as temperature and metallic content) with each detected target. The sequence of events that occurred between the Knowledge Assistant and its specialists is described below.

The Knowledge Assistant first receives the bounding box information (the volume that the operator would like to scan) from the world model. User-supplied information such the scan height and resolution information of each sensor are stored in the knowledge base of the Knowledge Assistant (Figure 6). Several rules in the knowledge base combine the sensor requirements to determine the robot path and speed (Figure 7). This information is passed back to the world model where the robot motion is graphically previewed by the operator. If accepted by the operator, the robot begins its path.

The raw sensor data along with the position and orientation of the robot is collected in files by the data acquisition specialist. The data-acquisition specialist can also send real-time data to other specialists, such as the statistical pattern-recognition specialist, for real-time classification. Live data can be displayed on strip charts and regional maps by the data-acquisition specialist if desired. When a robot pass is complete, the Knowledge Assistant sends the file information to the other specialists for analysis of the sensor data. Magnetometer and pyrometer data file names are sent to the MAPER specialist as soon as the Knowledge Assistant receives the data files names. The Knowledge Assistant receives statistical information from MAPER for each of the files, executes an additional set of rules, and sends threshold data back to MAPER to help guide its analysis. When the MAPER analysis is completed, target file names are received from MAPER. The Knowledge Assistant reads in a target file as an instance. The process is repeated as each robot pass is completed.



Sensors Available

- ☑ magnetometer
- ☑ pyrometer
- ☐ EM-61

Input Change in Height for Scans (inches)

Height Delta 6

Input Sampling Resolution

- ◇ 1 inch
- ◇ 2 inches
- ◆ 3 inches
- ◇ 4 inches
- ◇ 6 inches
- ◇ 0 overlap

Spot Size 0.5

Sampling Rate (Hz) 2

X1 2.0	Y1 2.0	Z1 2.0
X2 40.0	Y2 40.0	Z2 40.0

The Scan-Width 3

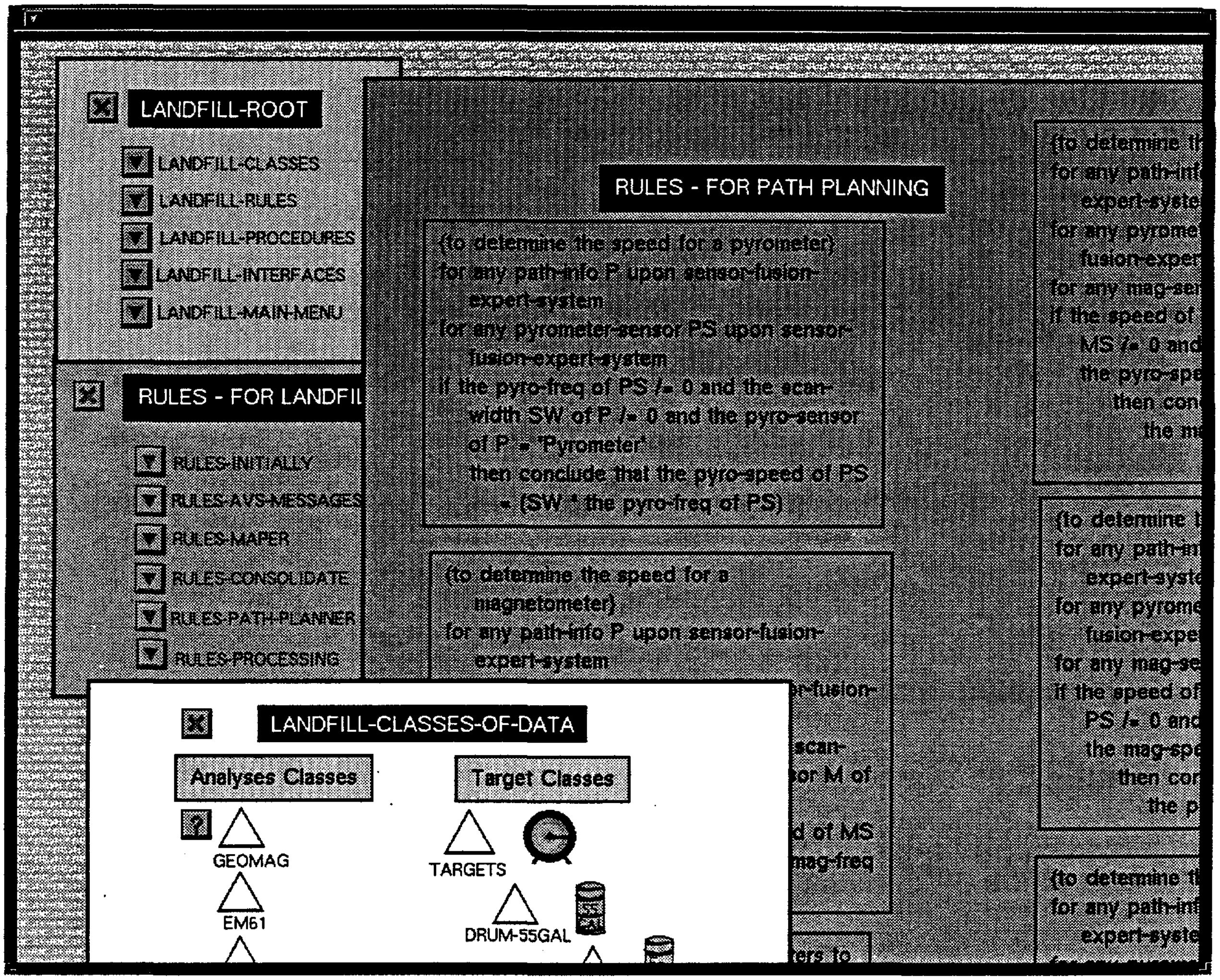
The Scan Height 2.0

The Scan Speed 6

The Scan Overlap 0

Figure 6. Expert system input worksheet and path information returned to robot controller.

When three robot passes have been completed and the file names have been received, the Knowledge Assistant sends the files names to the neural-net specialist. The neural-net specialist will analyze the data and return the target file to the Knowledge Assistant. The Knowledge Assistant reads the target file as an instance.



```
{to determine the speed for a pyrometer}
for any path-info P upon sensor-fusion-
    expert-system
for any pyrometer-sensor PS upon sensor-
    fusion-expert-system
if the pyro-freq of PS /= 0 and the scan-
    width SW of P /= 0 and the pyro-sensor
    of P = 'Pyrometer'
    then conclude that the pyro-speed of PS
        = (SW * the pyro-freq of PS)
```

```
{to determine the speed for a
    magnetometer}
for any path-info P upon sensor-fusion-
    expert-system
```

Figure 7. Sensor-specific rules for path planning.

When an instance of a target file is created and another instance of a target file already exists, a series of rules execute to consolidate the targets. A target is defined by x1, y1, z1, radius1, x2, y2, z2, radius2, magnetometer value, and pyrometer value. The target may not have radius values or have only one sensor magnitude value. The rules check each target instance attribute created by one file with each target instance attribute from the first target file. If any bounding boxes overlap or touch, the targets are combined into one target with an encompassing bounding box. If the bounding boxes do not overlap, the target attributes are added to the first target file. This process continues until all of the target files have been checked against the first target file. Reports are generated to convey the number of targets before and after consolidation. A file is then created by the Knowledge Assistant, which contains the consolidated target data. The file name is sent to the world model where the target objects and their attributes are displayed in a three-dimensional (3D) graphic environment.

The sequence of events described above was demonstrated at the 1995 DOE Industry/University/Lab Forum on Robotics for Environmental Restoration in Albuquerque, New Mexico. The Knowledge Assistant greatly simplified the operator's task by suggesting the initial scan path and by displaying the final data interpretation results in the 3D graphical environment of the world model. A distributed computing environment consisting of several UNIX workstations and a VME real-time computing system was used to implement the various components. While this made the system modular, it also complicated communications between the modules. An internally developed message passing scheme was implemented to handle these communications. In the future, communications between heterogeneous computers will be simplified with commercial CORBA (Common Object Request Broker Architecture) software.

For this first prototype, we also found that we did not use the forward and backward chaining capabilities of the expert system shell as much as initially anticipated. Many of the decisions were made in the form of independent "if-then" rules. If the sensors and tasks are well defined ahead of time, we would recommend not using a complex expert system such as G2, but to simply implement the rules in a programming language such as C or C++. However, if the sensor and tasks are not well defined, the expert system did provide considerable flexibility, and the inferencing mechanisms maybe useful for more complex tasks.

3.0 STATISTICAL PATTERN-RECOGNITION SPECIALIST

This section describes how statistical pattern-recognition techniques have been used to classify objects based on diverse sets of sensor readings. The same techniques have been used to discriminate between types of Underground Storage Tank (UST) waste and types of buried waste.

For the UST experiment, a Sandia-developed Mini-Lab sensor head (Figure 8) was mounted on a Fanuc S-800 robot arm. This sensor head contained a penetrometer, which measured viscosity, tip force, pore pressure, contact temperature, and pH. The sensor head also contained a spot pyrometer, Geiger counter, ultrasonic proximity sensors, camera, and gas-specific sensors. The robot arm was positioned above three barrels containing either supernate, sludge, or salt cake (actually simulated with water, Play-Doh, and pebbles). As the robot guided the sensor head into each of the barrels, the statistical pattern-recognition routines were used to determine if the penetrometer was in air, supernate, sludge, or salt cake. More importantly, if the computer can determine this, this information could be used to guide the robot arm so as not to push too vigorously into salt cake.

For the buried waste experiment, a sensor head containing a magnetometer and pyrometer was mounted on a Cimcorp gantry robot arm. This time, as the robot guided the sensor head over a mock buried waste site, the statistical pattern-recognition routines were used to classify the materials in real-time as metal, hot nonmetal, hot metal, and no target.

The statistical pattern recognition software is constructed to be used in two phases: During a learning phase, either simulated sensor readings or actual measurements from live sensors are used to learn the mean and variance of the sensor data in a particular class (e.g., metal, hot nonmetal, hot metal, and no target). During a follow-on monitoring phase, a multivariate, multiclass Bayesian classifier is used to identify the class. The results of the classifier are used by the expert system to notify the remote operator of the type of material sensed and any suggested plan of action (Figure 9). The minus-log-likelihood value indicates to the operator the probability of correct classification.

A statistical pattern-recognition technique, known as Bayesian hypothesis testing, is used to analyze sensor measurements and discriminate between classes. This method of classification is well recognized within the field of safeguards and nuclear material management [6,7,8]. Below is a brief introduction to the subject. A more detailed discussion can be found in Reference 9.

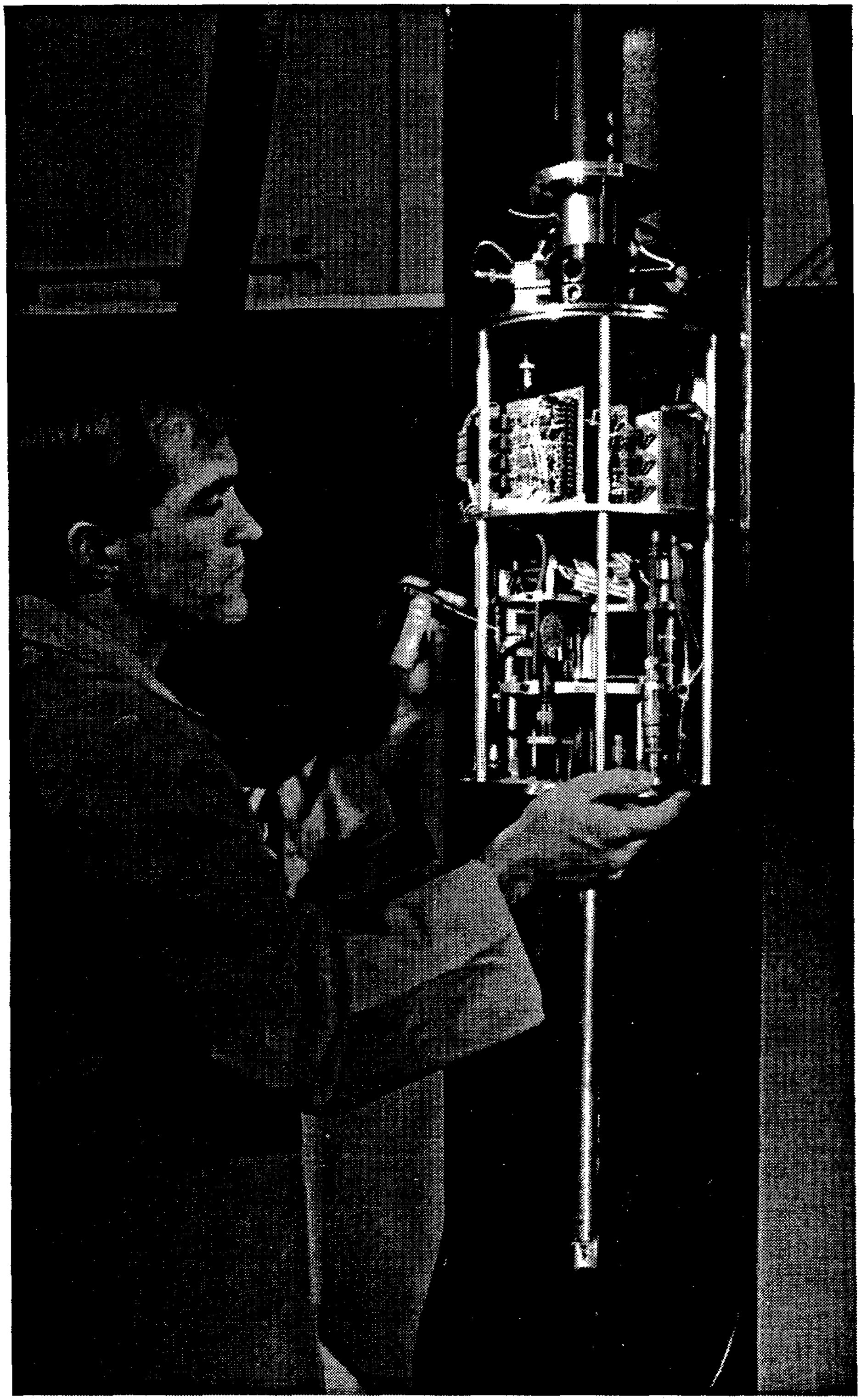
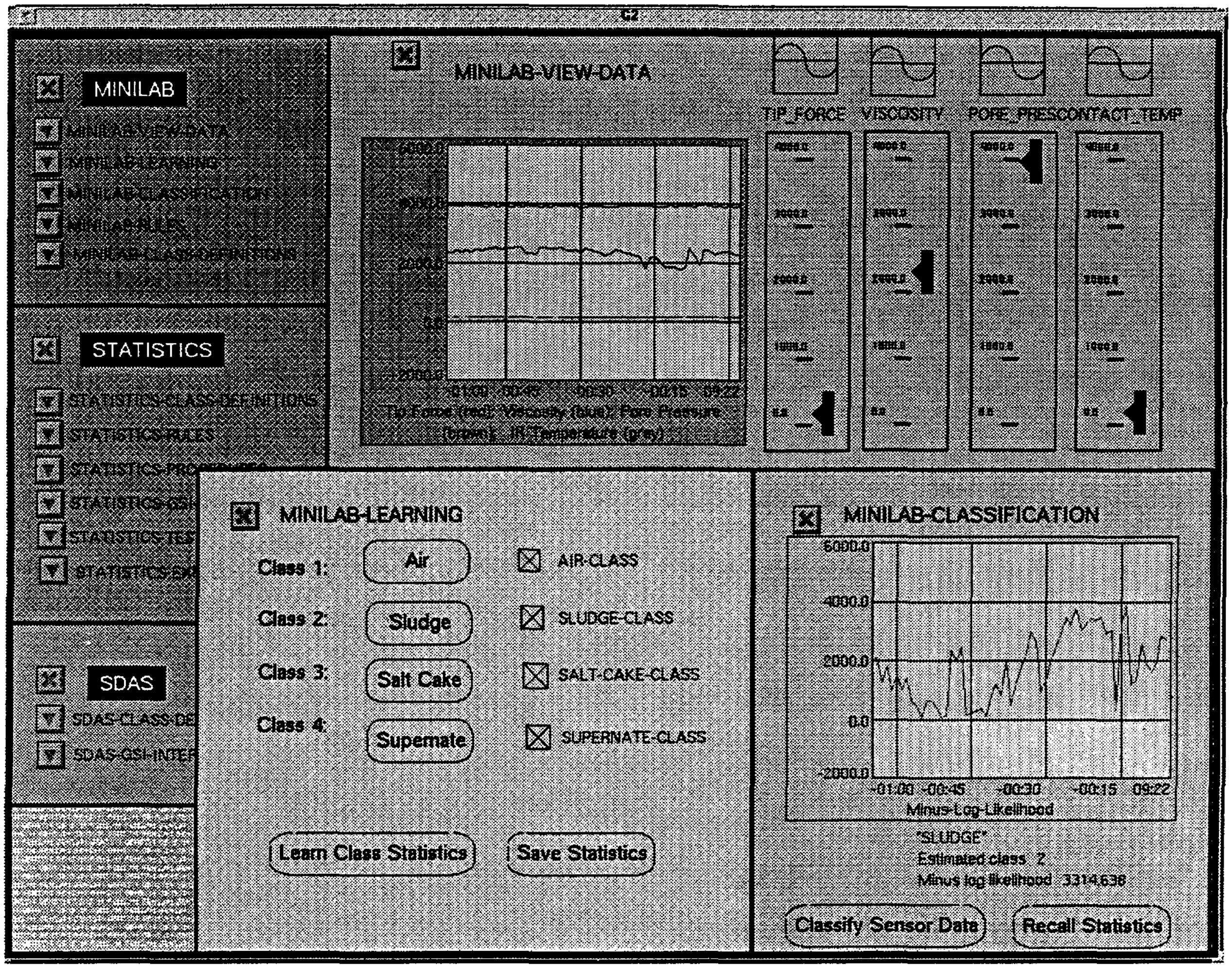


Figure 8. Mini-Lab sensor head used in the penetrometer experiments.



Class 1: Air ☒ AIR-CLASS

Class 2: Sludge ☒ SLUDGE-CLASS

Class 3: Salt Cake ☒ SALT-CAKE-CLASS

Class 4: Supernate ☒ SUPERNATE-CLASS

"SLUDGE"
Estimated class 2
Minus log likelihood 3314.638

Figure 9. Expert system user interface to the pattern-recognition specialist.

Bayesian hypothesis testing is used to classify a sample X of dimension n (i.e., $X \in \Re^n$) into one of m classes ω_i, $i = 1,\ldots,m$. The samples are assumed to come from a Gaussian (Normal) distribution with a mean and covariance that are learned prior to on-line testing. The decision rule for multiple classes is as follows: A sample X is an element of class ω_i if

$$P(\omega_i|X) > P(\omega_j|X) \qquad \forall j = 1,\ldots,m; j \neq i.$$

where $P(\omega_i|X)$ is the conditional probability that class ω_i occurred given sample X. The above equation selects the class that has the greatest probability.

Using Bayes' theorem, the conditional probability $P(\omega_i|X)$ can be replaced with $P(\omega_i)p(X|\omega_i)$ where $P(\omega_i)$ is the *a priori* probability that class ω_i will occur, and

$p(X|\omega_i)$ is the *a priori* conditional density function that sample X will occur if we know the class of X is ω_i. Assuming Gaussian distributions of X, the decision rule can be rewritten as: A sample X is an element of class ω_i if

$$h_{ij}(X) < \ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right] \qquad \forall j = 1,\ldots,m; j \neq i.$$

The minus-log-likelihood ratio is given by

$$h_{ij}(X) = 0.5(X-M_i)^t C_i^{-1}(X-M_i) - 0.5(X-M_j)^t C_j^{-1}(X-M_j) + 0.5\ln\left[\frac{\det(C_i)}{\det(C_j)}\right]$$

where M_i is the mean of X given class ω_i, and C_i is the covariance matrix of X given class ω_i. The superscripts *t* and *-1* denote matrix transpose and inverse operations. The function *det()* denotes the determinant of the matrix. This ratio is derived by taking the natural logarithm of the ratio of $p(X|\omega_j)$ to $p(X|\omega_i)$, both of which are normal distributions.

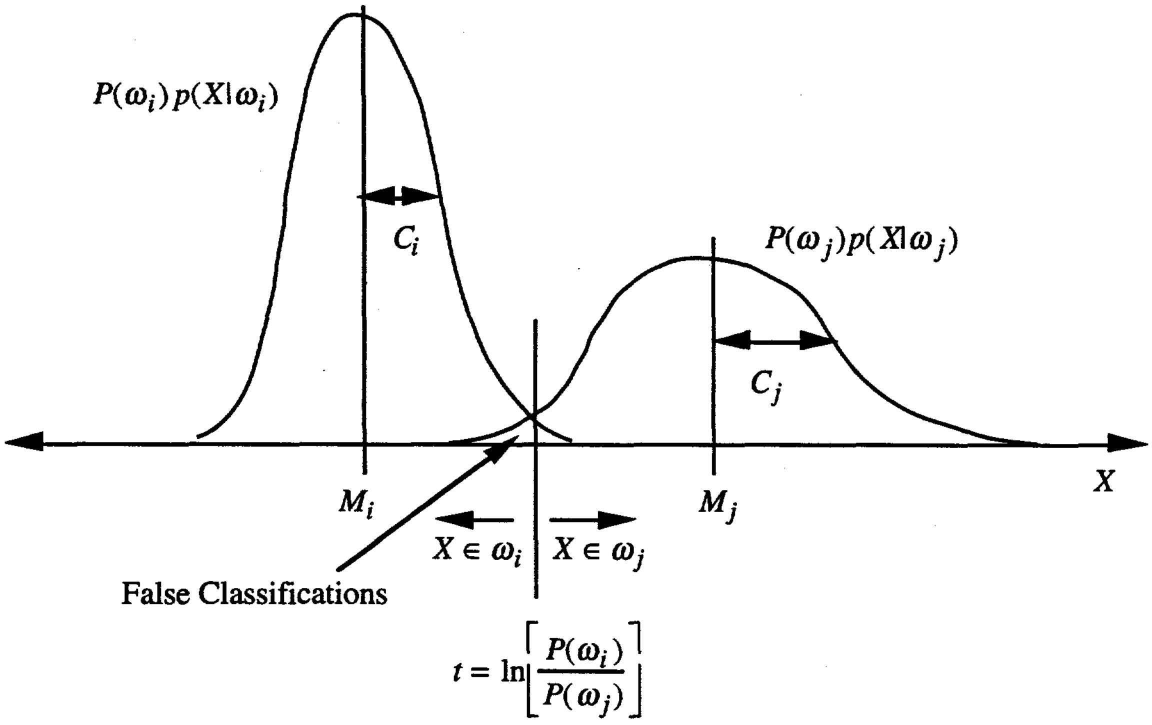

$P(\omega_i)p(X|\omega_i)$
C_i
$P(\omega_j)p(X|\omega_j)$
C_j
X
M_i
M_j
$X \in \omega_i$
$X \in \omega_j$
False Classifications
$t = \ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right]$

Figure 10. Univariate, two-class example of Bayesian hypothesis testing.

Figure 10 illustrates a univariate (in *X*), two-class example of Bayesian hypothesis testing. The area under each curve is equal to the *a priori* probability of class ω_i. The sum of these areas for all classes is one. The value t is the threshold below which we assign the sample to class ω_i, and above which we assign the sample to class ω_j. The area of overlap of the two curves is the probability of false classification. To achieve the best classification, it is desirable to minimize the probability of false classification. This usually means that you would like the means of the two classes to be far apart and the covariances to be small (i.e., small standard deviation). Since the probability of error is computed during the learning phase (when the means and covariances are learned), it may be used as a measure for choosing the sensors that provide the best classification.

The total probability of error between classes ω_i and ω_j can be estimated by

$$\varepsilon_{ij_{total}} = P(\omega_i)\varepsilon_{1ij} + P(\omega_j)\varepsilon_{2ij}$$

where

$$\varepsilon_{1ij} = 0.5 - erf\left(\frac{t_{ij} - \eta_{1ij}}{\sigma_{1ij}}\right) \qquad \text{and} \qquad \varepsilon_{2ij} = 0.5 - erf\left(\frac{\eta_{2ij} - t_{ij}}{\sigma_{2ij}}\right)$$

$$t_{ij} = \ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)$$

$$\eta_{1ij} = E\{h(X)|\omega_i\} = 0.5\sum_{l=1}^{n}\left[\left(1 - 1/\lambda_l\right) - \left(d_{jl} - d_{il}\right)^2/\lambda_l + \ln\left(1/\lambda_l\right)\right]$$

$$\eta_{2ij} = E\{h(X)|\omega_j\} = 0.5\sum_{l=1}^{n}\left[\left(\lambda_l - 1\right) - \left(d_{jl} - d_{il}\right)^2 + \ln\left(1/\lambda_l\right)\right]$$

$$\sigma_{1ij} = Var\{h(X)|\omega_i\} = 0.5\sum_{l=1}^{n}\left[\left(1 - 1/\lambda_l\right)^2 - \left(d_{jl} - d_{il}\right)^2/\lambda_l^2\right]$$

$$\sigma_{2ij} = Var\{h(X)|\omega_j\} = 0.5\sum_{l=1}^{n}\left[\left(\lambda_l - 1\right)^2 - \lambda_l\left(d_{jl} - d_{il}\right)^2\right]$$

The errors ε_{1ij} and ε_{2ij} are associated with incorrectly choosing ω_i when we should have chosen ω_j and vice versa. The means η_{1ij} and η_{2ij} are the conditional estimates of the minus-log-likelihood given either class ω_i or ω_j. The variances σ_{1ij} and σ_{2ij} are the conditional variance of the minus-log-likelihood given either class ω_i or ω_j. The values λ_l are the resulting diagonal elements of covariance

C_j after simultaneously diagonalizing it with C_i (i.e., find A such that $AC_iA^t = I$ and $AC_jA^t = \Lambda$ where Λ is a diagonal matrix with elements λ_l). The values d_{jl} and d_{il} are the resulting elements of the means AM_i and AM_j.

Note that the above equation is only an estimate of the probability of error since the minus-log-likelihood is not Normal but instead quadratic when $C_i \neq C_j$. Also, note that this equation provides a pair-wise comparison between classes. These values can be put in matrix form as shown below.

$$\begin{bmatrix} 0 & \varepsilon 12_{total} & \varepsilon 13_{total} & \varepsilon 14_{total} \\ \varepsilon 12_{total} & 0 & \varepsilon 23_{total} & \varepsilon 24_{total} \\ \varepsilon 13_{total} & \varepsilon 23_{total} & 0 & \varepsilon 34_{total} \\ \varepsilon 14_{total} & \varepsilon 24_{total} & \varepsilon 34_{total} & 0 \end{bmatrix}$$

By looking across a row, we can see which classes appear to overlap causing larger error terms. By adding up ith row, we can estimate the total error associated with choosing class ω_i.

Overall, we found that the statistical pattern recognition technique worked well if the sensor readings were repeatable and stable. For example, we had a very high detection rate using the magnetometer and pyrometer. Unfortunately, when measuring the simulated Underground Storage Tank waste, the penetrometer did not always return repeatable results. The viscosity measurement was different depending on the sequence of events (e.g., inserted in sludge and then salt cake, or first inserted in salt cake and then sludge). Also, since we taught the system based on the steady-state sensor readings, we were not able to capture transient responses such as ringing, which occurred when the penetrometer entered the salt cake. These two results imply that in the future we should add some time-dependent features (e.g., tip force readings at the current time and some delayed time) to the sample X.

Another problem with Bayesian classification is that there is no way to capture a sense of nonbelief. In other words, we should be able to say that I do not believe that it is any of the four classes, and therefore, it must be something else. In the future, we may be able to solve this by using Dempster-Shafer belief functions.

4.0 NEURAL-NETWORK SPECIALIST

In our tests, the goal of the neural-network specialist was to map the heat sources in an area by examining pyrometer data from multiple scans over the area. The pyrometer provides a measurement of the average temperature within its field of view. In our case, we used a pyrometer with a 2:1 field of view. This means that the spot diameter is one-half the distance between the pyrometer and the waste surface. Because of this averaging effect, the temperature map created by scanning the pyrometer over the surface becomes blurred as the distance between the pyrometer and the waste increases. The neural network takes the raw sensor maps at three heights and determines the number, location, size, height, and temperature of sources. This work is also applicable to other radiation sources, given suitable sensors.

The reason that a neural network was used for this task was to evaluate how learning algorithms could be used to analyze sensor readings over time. The hypothesis is that expert knowledge will be required to analyze the first few waste sites. However, learning algorithms will be useful as more information becomes available over time. Learning sets from previous sites will be used to train the neural networks. Eventually, the neural networks will be able to perform a higher-quality analysis of extremely complex, nonlinear, multivariate data.

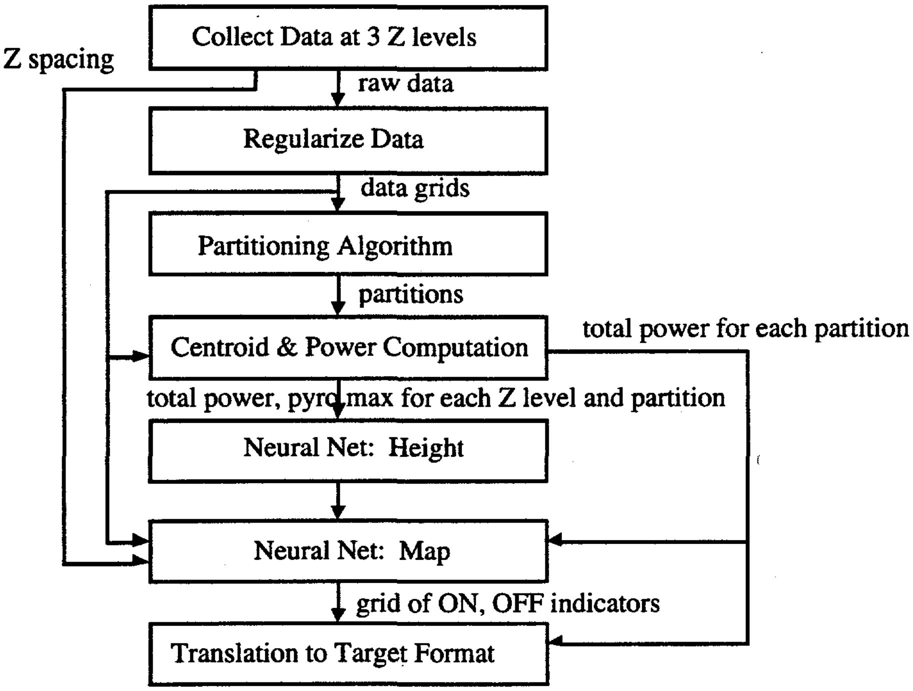

Collect Data at 3 Z levels
Z spacing
raw data
Regularize Data
data grids
Partitioning Algorithm
partitions
Centroid & Power Computation
total power for each partition
total power, pyro max for each Z level and partition
Neural Net: Height
Neural Net: Map
grid of ON, OFF indicators
Translation to Target Format

Figure 11. Flow diagram for neural-network processing of pyrometer data.

Figure 11 is a flow chart of the neural-network training process. During data collection, the sensing axis of the pyrometer is pointed straight down and is moved by the robot arm in a serpentine scan pattern over the selected area. The algorithms developed here require scans regularly spaced in Z (height). Temperature data is collected at regular time-sampling intervals; however, since the speed of the robot changes, the data is not collected on an evenly spaced grid. Therefore, the next step, "Regularize Data," creates a regularly spaced grid of pyrometer data for each Z level. Regularized data for one pyrometer scan is shown in Figure 12. White lines indicate edges of heat sources: a pipe wrapped with heat tape, and two rectangular heat sources at 400 degrees F and 250 degrees F. Three scans were taken 6 inches apart in height. The rectangular heat sources were 17.7 inches and 18.1 inches beneath the lowest scan plane.

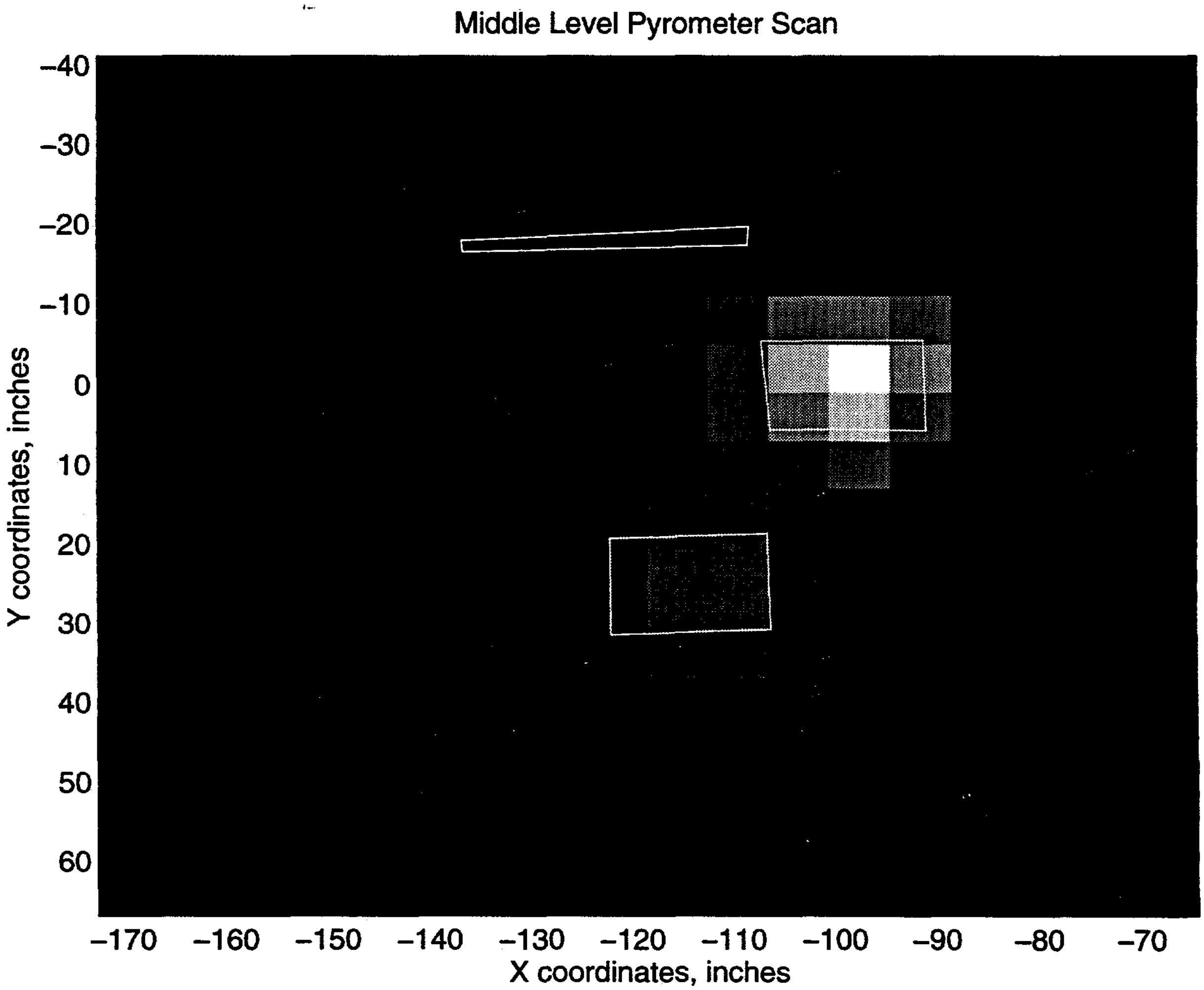


Figure 12. Middle-level pyrometer scan data.

Next, a partitioning algorithm is applied to the lowest-level scan. The number of sources are identified, and each square in the grid is assigned to a partition. Two partitions are identified, each encompassing one of the rectangular heat sources. The heated pipe did not generate sufficient signal for it to be picked out by the partitioning algorithm. The centroid and power computation algorithms are applied to each partition. The power is defined as the sum of the readings in the partition, subject to the accuracy of the partitioning algorithm and the inaccuracies caused by overlapping of source images. Note that this is not the radiant power, which is a function of temperature to the fourth power.

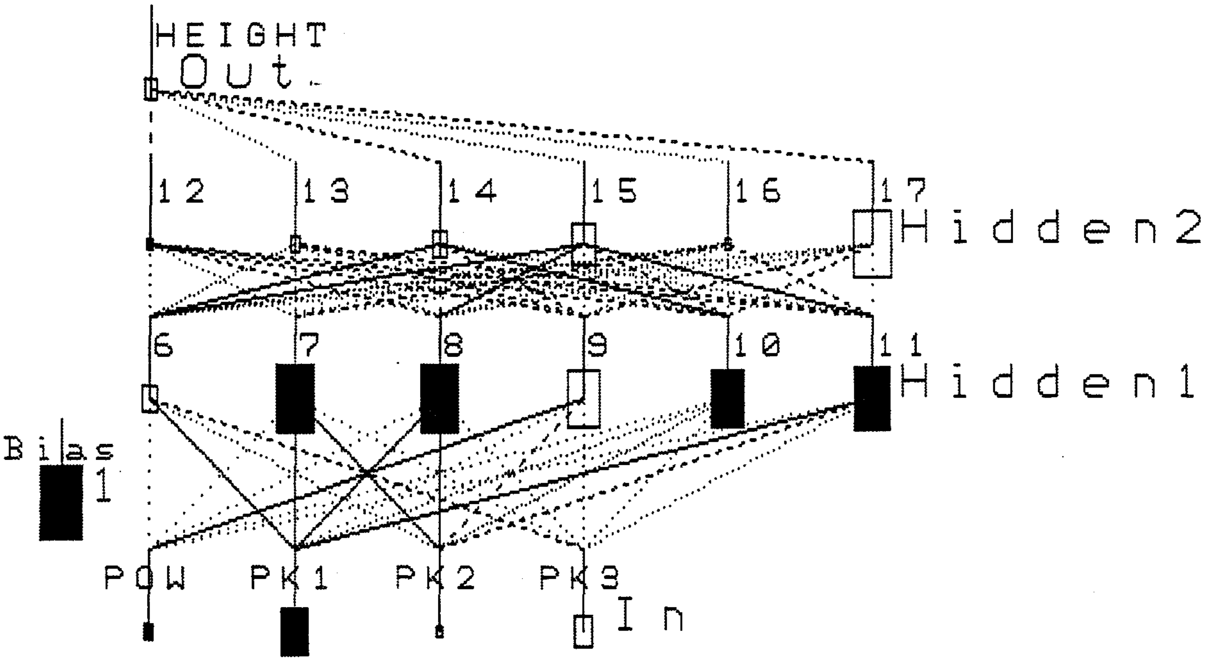


Figure 13. Height-determination network.

The height-determination network shown in Figure 13 is run once for each partition. This is a two-layer, fully connected, feedforward network that has been trained by a backpropagation algorithm [10]. The sizes of the boxes represent the last value through the network at that node, and whether the box is shaded or empty implies the value is positive or negative. The weights at each node are the result of the backpropagation technique. Inputs to this network are (a) power estimate for the partition (POW), and (b) peak pyrometer readings for each Z level for the partition (PK1, PK2, PK3). The network output is an estimate of the distance from the lowest measurement plane to the plane of the source.

Similarly, a mapping network is executed once for each cell in the X-Y grid. This is also a two-layer, fully connected, feedforward network that has been trained by a backpropagation algorithm. Inputs are (a) the distance between scans in the Z direction, (b) the height from the height-determination network, and (c) the pyrometer readings for this location on the grid for each scan. The network output is interpreted as a binary signal, 1 indicating presence of the source at this cell, and 0 indicating absence. The results on the example problem are shown in Figure 14. White cells indicate presence of a source, black indicates absence.

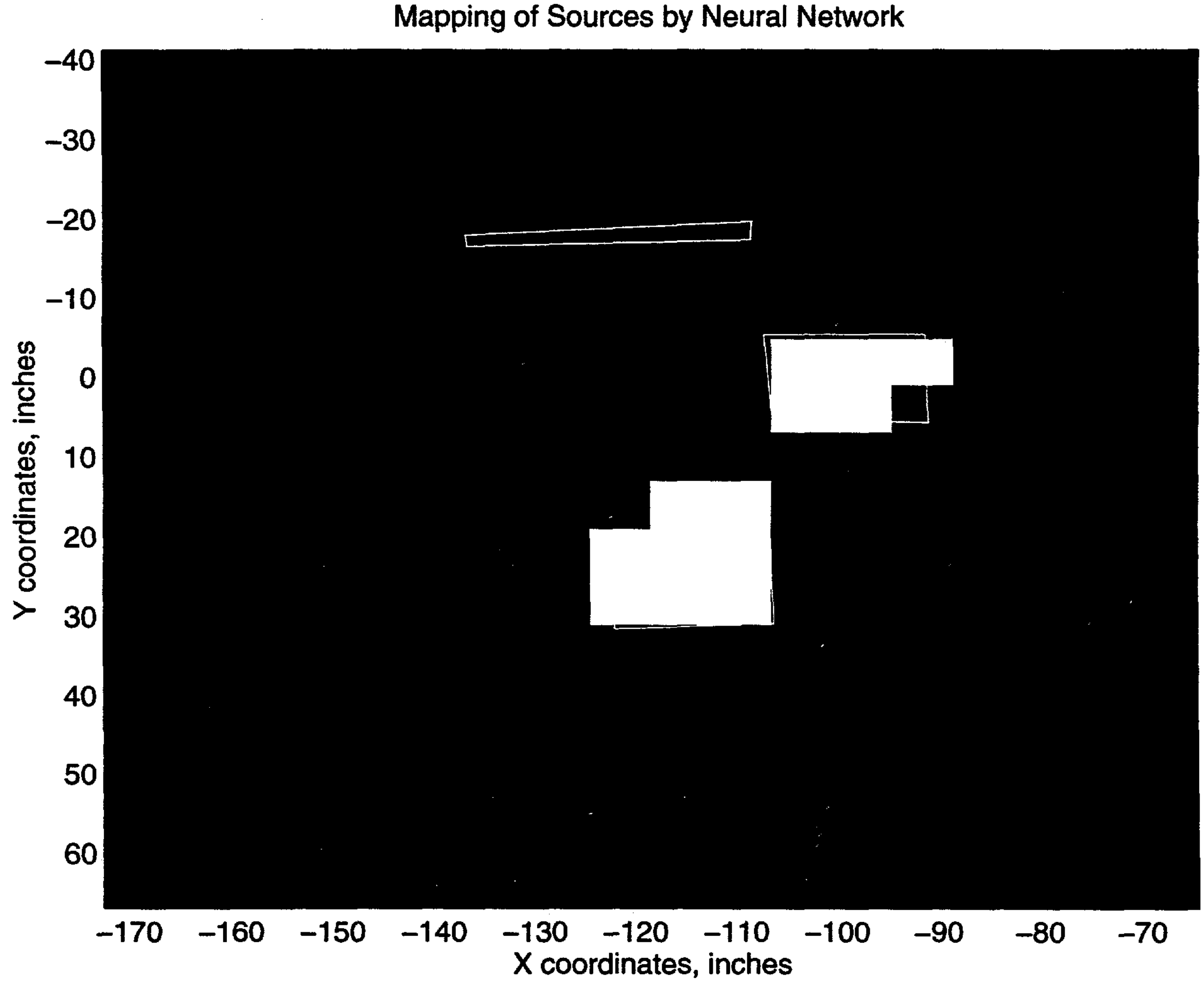


Figure 14. Mapping of sources by mapping network.

The last step of the process is to translate the results into the target file format, indicating the height of each source and its maximum extent in an X-Y coordinate system. The temperature of each source is the power estimate divided by the number of cells indicated by the mapping network (in other words, the average temperature measured per identified object).

A total of 12 experimental setups were created using a rectangular heat source, a light bulb, and the heat-taped pipe. Scans were taken at 4 Z levels in such a way that each experiment yielded two complete data sets. That is, one data set had Z spacing of 6 inches, while the other had a Z spacing of 12 inches. Thus there were effectively 24 experiments from which to create data.

The partitioning algorithm was developed using the first six experiments, leaving the remainder of the experiments for validation of the results. As noted above, the heat-taped pipe was not hot enough or big enough to show up in an experiment with the hotter sources.

The centroid and power computation algorithms first subtract off background level from the data grids, setting entries less than zero equal to zero. The centroid for each partition is thus computed using positive and zero values. The scale factor and offset for the power computation are determined experimentally using readings taken from the rectangular heat sources at different temperatures.

The height and mapping neural networks were trained by dividing the data into test and training data sets so that generalization could be examined. Performance on height network is tabulated in Table 1. Training was stopped when performance on the test set began to decline. The imperfect generalization seen here indicates that more training data would be needed to improve the accuracy of the network on non-training examples.

Table 1. Height network results.

	STD(error) (inches)	MIN(error) (inches)	MAX(error) (inches)
Training Set	1.13	-2.4	+1.5
Test set	2.18	-9.3	+3.8

The output of the mapping network is deemed to be 1 if the center of the cell is within the borders of a source. Hence some errors are to be expected in cells near the edges of a source. Performance on the mapping network is tabulated in Tables 2 and 3. Tabulated figures indicate the number of examples in each category. The training set was selected so that the numbers of 1 and 0 samples were equal, to keep the network from being biased toward zero valued outputs. This left an imbalance of 1 and 0 entries in the test set, as reflected in Table 3.

Table 2. Mapping network results on the training set.

	Actual 1 Output	Actual 0 outputs
Desired 1 Output	1919	127
Desired 0 Output	142	1931

Table 3. Mapping network results on the test set.

	Actual 1 Output	Actual 0 outputs
Desired 1 Output	199	17
Desired 0 Output	103	4943

It is easy to see that the approach to mapping here is limited compared to a true deconvolution algorithm. There are several dependencies (such as the height computation depending on the output of the power estimate) that contribute to a cumulative buildup of error as data progresses through the flowchart. Furthermore, the mapping step of this approach ignores the fact that each reading depends not only on the contribution of the source directly beneath the sensor, but also on the contribution of everything within the sensors' field of view.

It seems reasonable that a neural-network deconvolution algorithm could be written to better handle these difficulties. That algorithm would consist of a recurrent algorithm that took as inputs both sensor readings and previous outputs of the cell and its neighbors. Investigative work was done into this approach, but due to time considerations it was not adopted as the final approach.

5.0 EXPERT SYSTEM ASSESSMENT OF GAS CHROMATOGRAPHY DATA

This section describes our éfforts in developing an expert system for data assessment of gas chromatography results. Data assessment is the evaluation of data simply to see if the data is of acceptable quality for further analysis. Being able to assess data on-line and warn the operator of faulty equipment or equipment settings can result in substantial time and cost savings. The system developed here was designed to be fully automated, providing the operator with a list of symptoms and possible causes of equipment failure.

This system was developed in conjunction with an ongoing effort at Los Alamos and Sandia National Laboratories as part of their Contaminant Analysis Automation (CAA) project. The project incorporates the use of Standard Laboratory Modules (SLMs) and self-contained analysis procedures combined in an analysis pathway. This involves a series of SLMs, which are successively performed on a sample by inserting the sample into each SLM in turn, as needed, and then moving between SLMs with a robot arm. This fully automated process is intended to include: data assessment, quality-control analysis, and data-interpretation components. The work described in this report falls into the data-assessment category. The system would include functionality to monitor the progress of the analyses, and determine if any problems or unusual circumstances warranted stopping or changing the automated process.

The goal of the system is to take the raw gas chromatography data directly from the gas chromatography workstation, and create a file that is automatically sent to our data-assessment system. The data is then analyzed by procedures that will be described in detail later. The ultimate goal is to determine if the data is of acceptable or unacceptable quality. If unacceptable, we would like the system to determine the problem(s) and recommend solutions to remedy the problem. Where possible, these solutions could be automatically implemented, or if not, then an appropriate operator would be notified about the problem and recommended solution. The system would also stop the automated processing if additional analyses could not be performed properly.

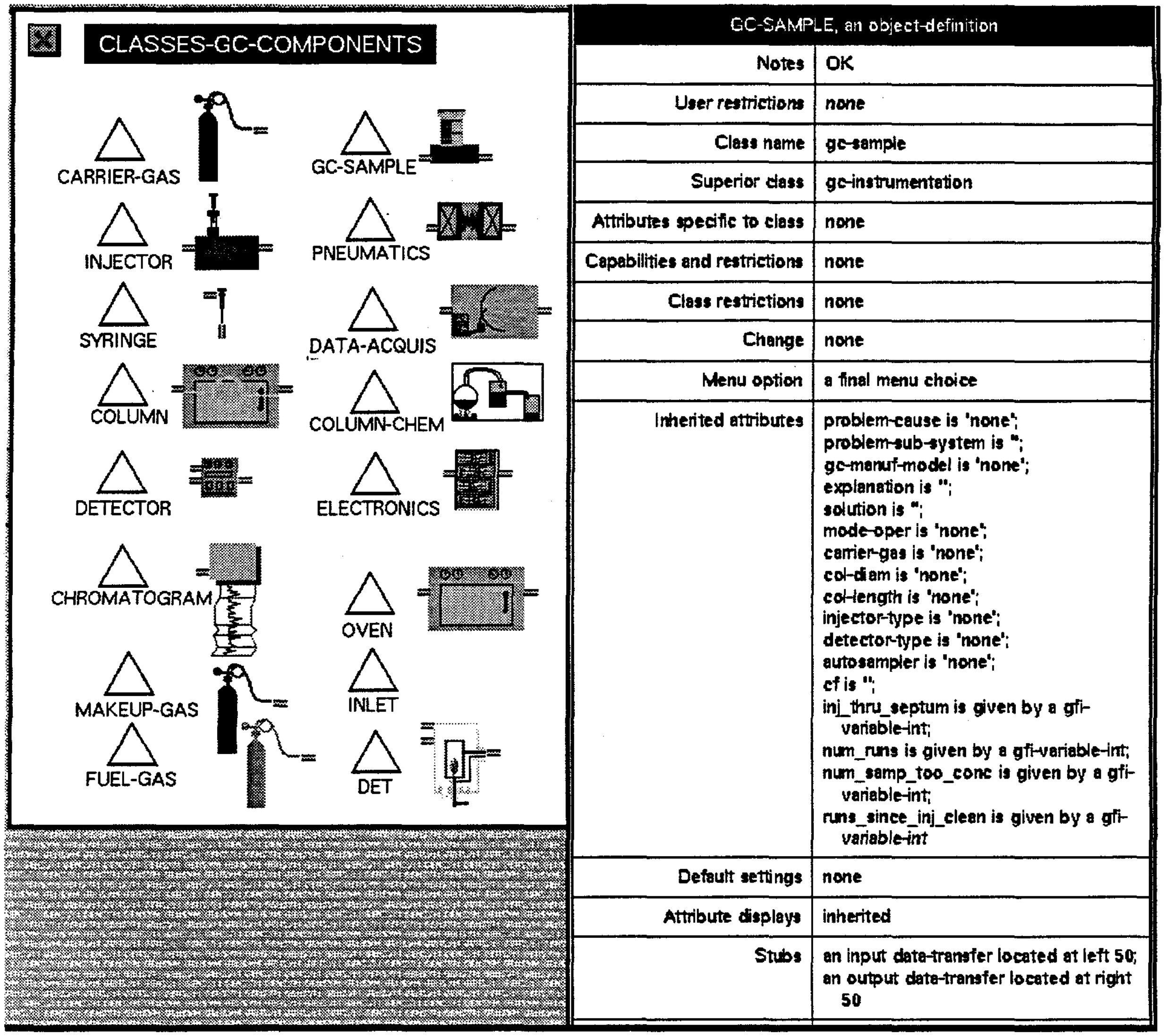
The implemented methodology involved generating gas chromatograms using the vendor equipment and software package (Varian GC3400 with Star software on a PC). A routine reformatted the data into NETCDF format and sent it to Matlab for signal-to-symptom processing. This processing took the raw data and used algorithms developed by Lós Alamos National Laboratory and Sandia National Laboratories to find symptoms relating to peak shape and location, and baseline trends and characteristics. The methodology used in the signal-to-symptom processing are described in References 11 and 12. This data was put into a "symptom file" and sent to the expert system for processing. The entire procedure is shown below:

1) GC --> Star Workstation --> PC --> NETCDF file;
2) Send file to Matlab for signal-to-symptom processing;
3) Generate Symptom file for input into expert system (via Message Server/FDB);
4) Input data into expert system, determine if data is of acceptable quality, or determine problem and recommend solution.

The method of assessing the data involved starting up the expert system along with a message server and connecting communication bridge. At that point, the expert system was "listening" for messages addressed to it. Whenever a symptom file was generated, a message containing the name of the file was sent to the expert system. Rules were set up that automatically loaded this file name into a scrollable menu, which listed all GC symptom files that had been received since the expert system started up. This file may be automatically or manually selected for input into the system and subsequent processing.

The symptom file was a specially formatted file (as required by the expert system), which contained a series of attributes and associated fuzzy values corresponding to the presence / absence / severity of each symptom. Each symptom was a feature related to the chromatogram signal, such as baseline trend, peak shape, peak retention time, etc. Examples of symptoms are peak tailing or leading peaks, where the shape of the peak deviates from a normal Gaussian distribution and is skewed one way or another. Other symptoms are "ghost" or extra peaks, changes in the expected retention time for a given peak, and drifting or changes in the baseline trend. Expert chemists have determined that there is a fixed relationship between the presence of particular subsets of these symptoms with particular problems with the gas chromatography instruments for a given analysis method. The symptom file contains fuzzy values ranging from -1.00 (completely absent/false symptom) to 1.00 (completely present/true symptom) for each symptom used in the system. It additionally includes some general header information identifying the source of the data, etc.

The expert system maintains an object-oriented view of the various components of the gas chromatography instrument. Figure 15 shows a series of objects that were created using G2's icon editor to represent various objects in the system. Also shown is a table of attributes for one of the objects. An input screen allows the user to enter the specific gas chromatography instrumentation details of his/her specific system (Figure 16). It contains information such as the type of instrument, mode of operation, which carrier gas is being used, column length and diameter, detector type being used, injector type being used, and whether an auto sampler is included. For example, if the detector type was not flame ionization, then the rule pertaining to "Dust in the flame" would not be applicable. This information allows this type of inferencing to be performed.

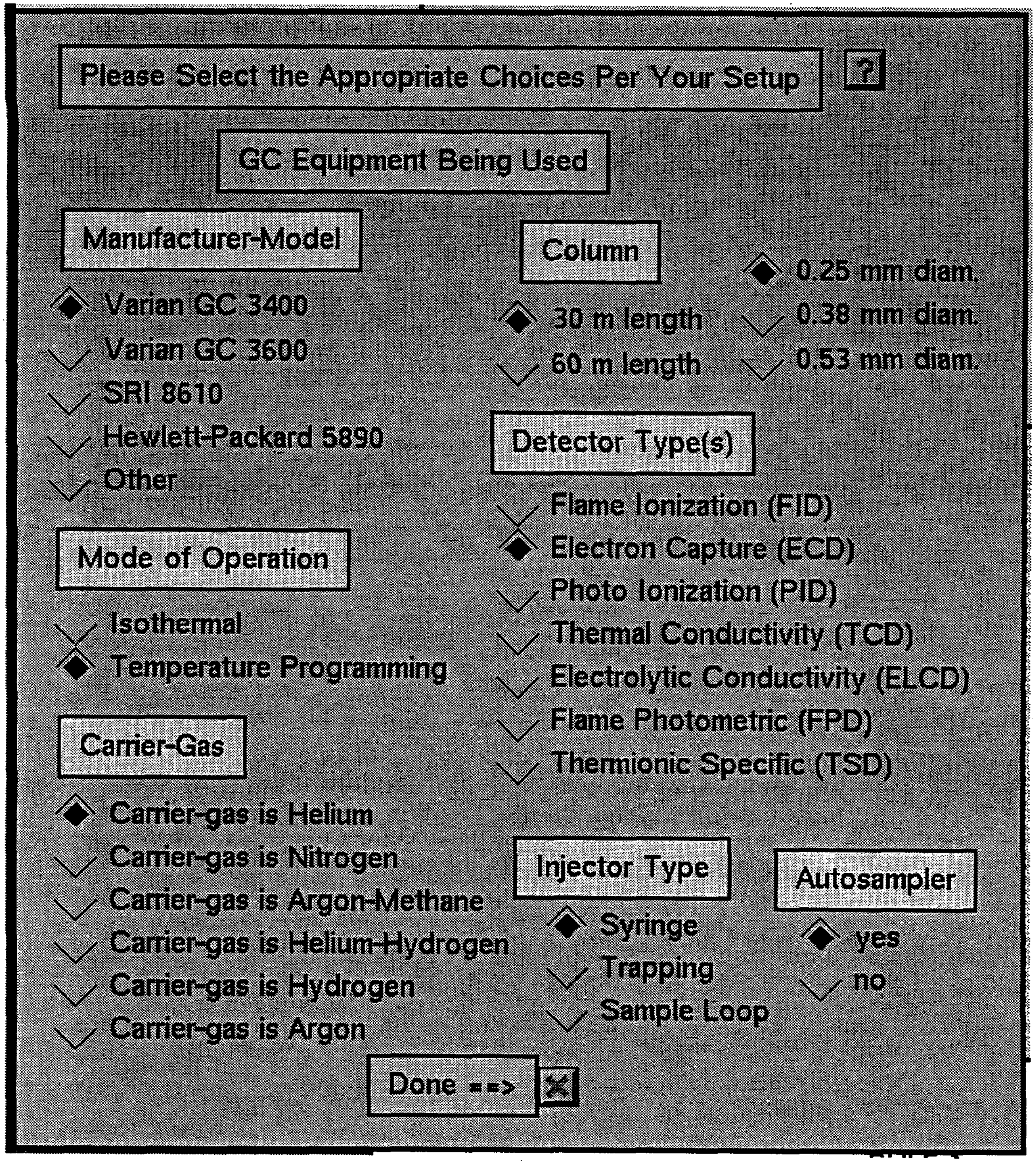


GC-SAMPLE, an object-definition	
Notes	OK
User restrictions	none
Class name	gc-sample
Superior class	gc-instrumentation
Attributes specific to class	none
Capabilities and restrictions	none
Class restrictions	none
Change	none
Menu option	a final menu choice

Inherited attributes	problem-cause is 'none';
	problem-sub-system is "";
	gc-manuf-model is 'none';
	explanation is "";
	solution is "";
	mode-oper is 'none';
	carrier-gas is 'none';
	col-diam is 'none';
	col-length is 'none';
	injector-type is 'none';
	detector-type is 'none';
	autosampler is 'none';
	cf is "";
	inj_thru_septum is given by a gfi-variable-int;
	num_runs is given by a gfi-variable-int;
	num_samp_too_conc is given by a gfi-variable-int;
	runs_since_inj_clean is given by a gfi-variable-int

Default settings	none
Attribute displays	inherited
Stubs	an input data-transfer located at left 50; an output data-transfer located at right 50

Figure 15. Example objects in G2 with attribute table.



Please Select the Appropriate Choices Per Your Setup [?]

GC Equipment Being Used

Manufacturer-Model

● Varian GC 3400
○ Varian GC 3600
○ SRI 8610
○ Hewlett-Packard 5890
○ Other

Column

● 30 m length
○ 60 m length

● 0.25 mm diam.
○ 0.38 mm diam.
○ 0.53 mm diam.

Mode of Operation

○ Isothermal
● Temperature Programming

Detector Type(s)

○ Flame Ionization (FID)
● Electron Capture (ECD)
○ Photo Ionization (PID)
○ Thermal Conductivity (TCD)
○ Electrolytic Conductivity (ELCD)
○ Flame Photometric (FPD)
○ Thermionic Specific (TSD)

Carrier-Gas

● Carrier-gas is Helium
○ Carrier-gas is Nitrogen
○ Carrier-gas is Argon-Methane
○ Carrier-gas is Helium-Hydrogen
○ Carrier-gas is Hydrogen
○ Carrier-gas is Argon

Injector Type

● Syringe
○ Trapping
○ Sample Loop

Autosampler

● yes
○ no

Done ==> ☒

Figure 16. User input form for gas chromatography instrumentation.

Whenever the expert system receives notification that a new GC symptom file has been received, it automatically imports the given file (assuming the automatic file input rule is enabled) and dynamically creates several objects (GC-chromatogram, GC-peak, and GC-instrumentation), which contain attributes corresponding to the values of the symptoms. A series of rules are in place that link the subset of symptoms to their possible causes. These rules are fired or activated if the appropriate symptom values are all above the required thresholds for the given rule. These rules associate each subset of symptoms with a given gas chromatogram instrument problem. The expert system has also subdivided a GC instrument into a series of subsystems. The problem(s) is determined and isolated as to particular subsystem. This should be helpful in trying to correct the problem. An example of one of these problem-determination rules is as follows:

> for any gc-peak GC that is part-of any gc-chromatogram that is generated-by any gc-instrumentation GI
> **if** the SensitivityChange of GC >= 0.5 and the RetentionTimeShift of the GC >= 0.5 and the GhostPeaks of the GC >= 0.5 and the Tailing of the GC <= 0.5
> **then** conclude that the problem-cause of GI = "Leaking at septum" and conclude that the problem-subsystem of GI = "injection."

A similar example rule is shown in Figure 17.

The system uses goal-driven (forward) chaining with a separate rule for each potential problem cause. Rules are written to ensure that causes with "Always" symptoms are only indicated if that symptom is present. Similarly, causes with "Never" symptoms will not be indicated if that symptom is present. Each rule includes confidence-factor algebra in which confidence factors are based on the expected versus actual presence of each possible symptom for a given cause. For each symptom that is present, if the fuzzy value is above an indicated threshold (reflecting its presence), then the expert system uses the raw value, and then multiplies that value by an "expected frequency factor." That "expected frequency factor " is:

Always = 1.0,
Usually = 0.75
Sometimes = 0.5
Infrequently = 0.25
Never = 0.0.

If the value is less than the threshold, then it is set to a value of "0," reflecting its absence. The values are then normalized to the range 0 to 1 to offset the differing number of symptoms present for each cause. Thus the confidence factor is the observed value divided by the total possible where

Total possible = (symptom[1] * expected frequency + symptom[2] * expected frequency + ... + symptom[n] * expected frequency).

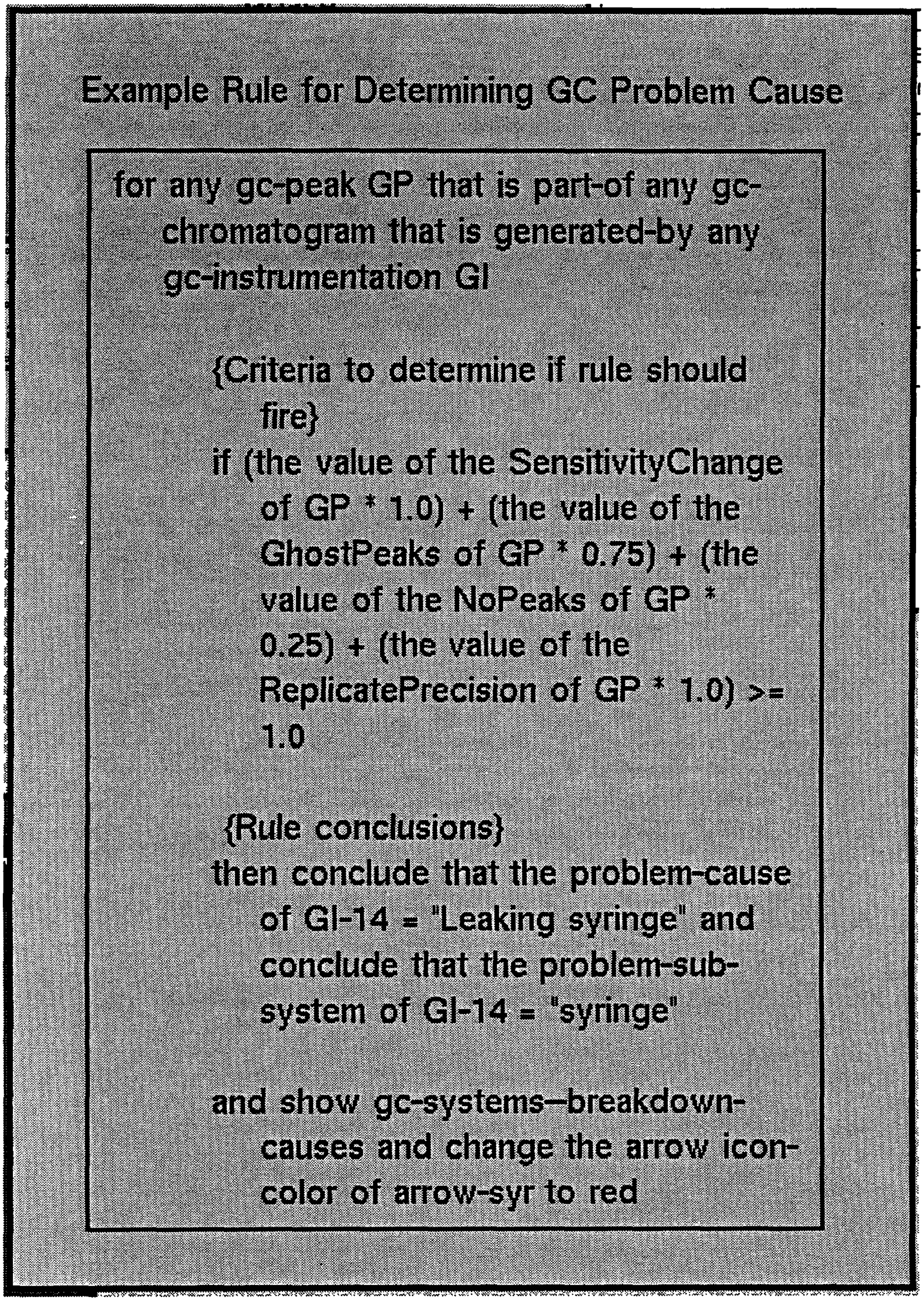
Figure 18 displays a selection of example rules, including one where the problem cause for "Dirt in syringe" is identified.

Current rules are based on a table of symptom/cause relationships which is modified from: Stillman & Du [13] , and Varian Corporation Guide to Diagnosis/Troubleshooting [14], with additional input from Elling [15], Robinson [16], and Thornberg [17] (Table 4).

When the inferencing is complete, a graphical schematic of the GC instrument is displayed with an arrow indicating the particular subsystem that contains the problem, as well as the specific problem and recommended solution. An additional workspace may be displayed that indicates which symptoms were used to determine the problem and what their observed values were. This helps answer a "why" query, which is information commonly requested by the user. If no problems are indicated, then the graphical workspace indicates that the GC is OK, or that the expert system could not match the symptoms with any problem cause in the current set of rules. Figure 19 shows the schematic of the gas chromatograph indicating problems that have been identified, with confidence factors and explanation of the most likely cause displayed.

The gas chromatograph expert system was demonstrated at the 1994 DOE Industry/University/Lab Forum on Robotics for Environmental Restoration in Albuquerque, New Mexico, and has been integrated into the CAA system as a standard laboratory module. Similar to the statements in Section 2.0, we found that for a simple set of rules relating symptoms vs. causes that it would be easier to implement a set of “if-then” rules in a common programming language. However, given a set of intertwined rules, the forward and backward chaining of the expert system may provide better results. How good these results will be depends on the Knowledge Engineer who develops the rules.

Finally, an extremely useful feature, which we rather crudely implemented in G2, is the ability to trace the rules fired by the inference engine and provide the operator with not only the end conclusion, but also how it came to this conclusion. Future expert systems used for these types of analyses should have this feature built into the shell.



Example Rule for Determining GC Problem Cause

for any gc-peak GP that is part-of any gc-chromatogram that is generated-by any gc-instrumentation GI

{Criteria to determine if rule should fire}

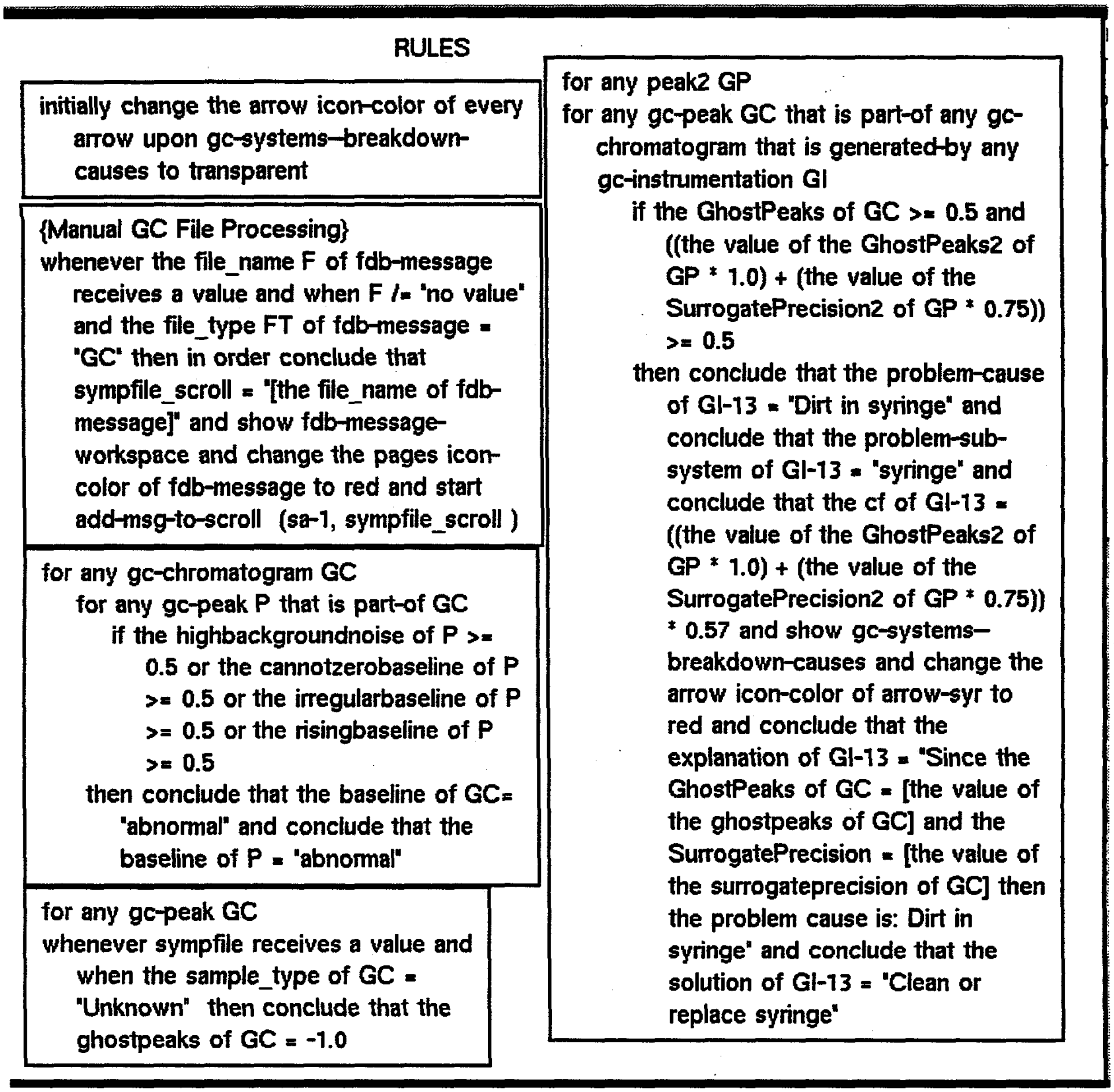
if (the value of the SensitivityChange of GP * 1.0) + (the value of the GhostPeaks of GP * 0.75) + (the value of the NoPeaks of GP * 0.25) + (the value of the ReplicatePrecision of GP * 1.0) >= 1.0

{Rule conclusions}

then conclude that the problem-cause of GI-14 = "Leaking syringe" and conclude that the problem-subsystem of GI-14 = "syringe"

and show gc-systems—breakdown-causes and change the arrow icon-color of arrow-syr to red

Figure 17. Example rule for problem cause determination.



RULES

```
initially change the arrow icon-color of every
   arrow upon gc-systems–breakdown-
   causes to transparent
```

```
{Manual GC File Processing}
whenever the file_name F of fdb-message
   receives a value and when F /= 'no value'
   and the file_type FT of fdb-message =
   'GC' then in order conclude that
   sympfile_scroll = "[the file_name of fdb-
   message]" and show fdb-message-
   workspace and change the pages icon-
   color of fdb-message to red and start
   add-msg-to-scroll  (sa-1, sympfile_scroll )
```

```
for any gc-chromatogram GC
   for any gc-peak P that is part-of GC
      if the highbackgroundnoise of P >=
         0.5 or the cannotzerobaseline of P
         >= 0.5 or the irregularbaseline of P
         >= 0.5 or the risingbaseline of P
         >= 0.5
     then conclude that the baseline of GC=
         'abnormal' and conclude that the
         baseline of P = 'abnomal'
```

```
for any gc-peak GC
whenever sympfile receives a value and
   when the sample_type of GC =
   'Unknown'  then conclude that the
   ghostpeaks of GC = -1.0
```

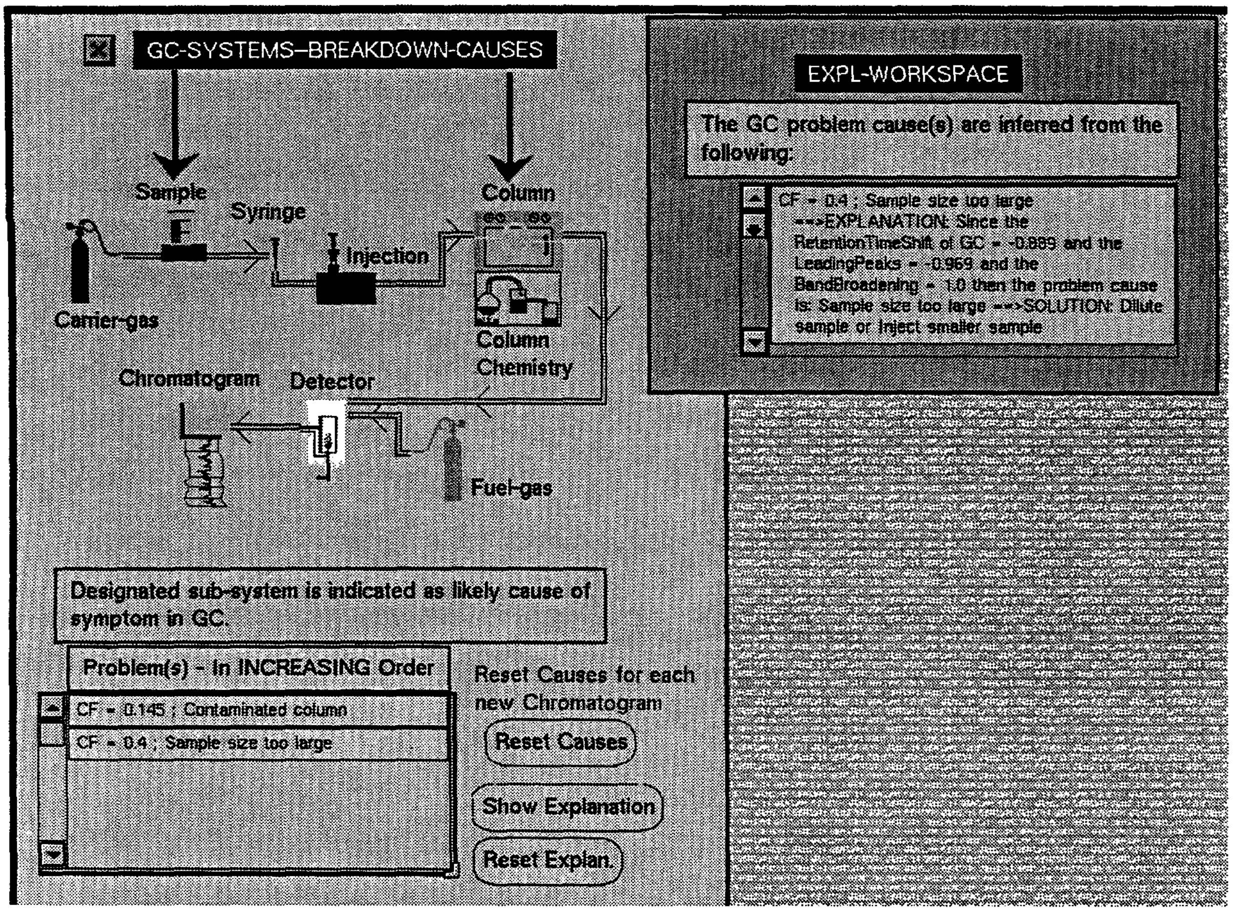
```
for any peak2 GP
for any gc-peak GC that is part-of any gc-
   chromatogram that is generated-by any
   gc-instrumentation GI
      if the GhostPeaks of GC >= 0.5 and
         ((the value of the GhostPeaks2 of
         GP * 1.0) + (the value of the
         SurrogatePrecision2 of GP * 0.75))
         >= 0.5
      then conclude that the problem-cause
         of GI-13 = 'Dirt in syringe' and
         conclude that the problem-sub-
         system of GI-13 = 'syringe' and
         conclude that the cf of GI-13 =
         ((the value of the GhostPeaks2 of
         GP * 1.0) + (the value of the
         SurrogatePrecision2 of GP * 0.75))
         * 0.57 and show gc-systems–
         breakdown-causes and change the
         arrow icon-color of arrow-syr to
         red and conclude that the
         explanation of GI-13 = "Since the
         GhostPeaks of GC = [the value of
         the ghostpeaks of GC] and the
         SurrogatePrecision = [the value of
         the surrogateprecision of GC] then
         the problem cause is: Dirt in
         syringe" and conclude that the
         solution of GI-13 = 'Clean or
         replace syringe'
```

Figure 18. Example expert system rules.

	Leaking Syringe	Dirt in Syringe	Sample Decomposition	Late Elution	Column Bleed	Contaminated Column	Contaminated Detector	Wrong Fuel Gas (FID only)	Dust in Flame (FID only)	Make-up gas (ECD only)	Dirt in Injector	Leaking Septum	Sample too Concentrated	Sample Size too Large	Carrier Gas Low	Column Degradation
Retention time change				U	N						I	S	S	N	A	A
Spike precision and reproducibility		U	S													
Sensitivity change	A				N		A	A		A	S	A			U	
Peak tailing						S				S	S	N				S
Unresolved peaks										S		S	S		S	S
Band broadening				A									A	S	S	
Clipped peaks										I			A			
Rounded peaks										I			U			
Negative dip after peak (ECD only)						S	U									
Baseline drift - irregular						S	U					S				
Baseline drift - rising					A	U										
Clipped baseline					S		S									
High background noise						U	U				U					
Irregular spikes									A							
Ghost / Extra peaks	U	A	U	A		S					S	S	S			
No peaks	I															
Replicate precison and reproducibility	A		U													
Leading peaks			S											U	I	S

Key: A= Always, U= Usually, S= Sometimes. I= Infrequently, N= Never

Table 4. Symptoms vs. causes for gas chromatography.



The GC problem cause(s) are inferred from the following:

CF = 0.4 ; Sample size too large ==>EXPLANATION: Since the RetentionTimeShift of GC = -0.889 and the LeadingPeaks = -0.969 and the BandBroadening = 1.0 then the problem cause is: Sample size too large ==>SOLUTION: Dilute sample or Inject smaller sample

Designated sub-system is indicated as likely cause of symptom in GC.

Problem(s) - In INCREASING Order

CF = 0.145 ; Contaminated column

CF = 0.4 ; Sample size too large

Reset Causes for each new Chromatogram

Figure 19. Gas chromatograph schematic indicating problem causes and explanation.

6.0 CONCLUSIONS AND FUTURE DIRECTIONS

The prototype systems described here are examples of the type of automated processing that are desirable for future work in the environmental arena. The combination of automated processing together with intelligent analysis algorithms and a friendly graphical user interface provide a valuable means of improving and speeding up environmental analyses. The expert system provides a high-level means to control processing, with potential for future incorporation of additional sensors, further processing, and additional inferencing to determine more accurate results.

We believe that the combination of expert system technology along with statistical hypothesis testing and neural networks provides a valuable means of fusing sensor data. The Bayesian hypothesis testing provides an analytical means of classifying complex multivariate data, which would otherwise be near impossible to analyze by hand. It can also be used to rapidly select and test different sensor combinations for different characterization tasks. A neural network can be used to accurately model nonlinear, complex relationships. The expert system provides a high-level user interface, which can be used to alert operators and inform them of the appropriate next steps. As used in the gas chromatography data assessment, the expert system is invaluable as a real-time trouble-shooting guide for instrument failure.

Future work should focus on the area of data fusion and intelligent inferencing methodologies. This involves incorporating multiple raw data types into a single analysis in order to make better inferences and determine more information than is possible with any single raw data type alone. Goals for this processing include: more accurate location solutions, determination of target material type, and identification of target (i.e., 55-gallon steel drum), determination of container status (leaking, intact), container contents (or lack thereof), etc. This will allow future clean-up operations to be performed more safely, more efficiently, and with better results.

REFERENCES

[1] Feddema, J.T., "A Miniaturized Sensor System for In Situ Robotic Characterization of Hazardous Waste," *SPECTRUM '92*, Boise, Idaho, August 23-27, 1992, pp. 75-80.

[2] Feddema, J.T., and Spletzer, B.L., "Generic Data Acquisition System for Robotic Waste Characterization," Proceedings of Space 94, Albuquerque, NM, February 26 to March 3, 1994.

[3] Hensel, E., et al., "Subsurface Characterization Using Geophysical Methods," *International Meeting on Nuclear nad Hazardous Waste Management: SPECTRUM 94*, August 1994.

[4] Macy, J.A., *Infrared Pyrometry for Waste Characterization*, Masters Thesis, Department of Mechanical Engineering, New Mexico State University, December 1995.

[5] Gensym Corporation, *G2 Reference Manual*, Version 3.0, 125 Cambridge Park Drive, MA 02140, (617)547-2500.

[6] Avenhaus, Rudolf, *Safeguards Systems Analysis*, Plenum Press, New York and London, 1986.

[7] Bowen, W. Michael, and Bennett, Carl A., *Statistical Methods for Nuclear Material Management*, NUREG/CR-4604, The Superintendent of Documents, U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20013-7082.

[8] Argentesi, F., Avenhaus, R., Franklin, M., and Shipley, J.P., *Mathematical and Statistical Methods in Nuclear Safeguards*, Hardwood Academic Publishers, New York, 1983.

[9] Fukunaga, Keinosuke, *Introduction to Statistical Pattern Recognition*, Academic Press, New York and London, 1972.

[10] NeuralWare, Inc, NeuralWorks Professional II/Plus and NeuralWorks Explorer Reference Guide, Version 5.0, Penn Center West, Building IV, Pittsburgh, PA 15276, (412) 787-8222.

[11] Roberts, R.S. ad Bracht, R.R., "Quality Assessment of Gas Gromatorgrams, FY 1994 LANL CC&AT Final Report," LA-UR-94-3541, October 1994.

[12] Roberts, R.S. and Lahiri, S., "Chromatogram Assessment Project, FY 1995 LANL CC&AT Progress Report," February 1995.

[13] Stillman, M. and Du, H., "Automatic Knowledge Based Generation for GC diagnosis System," In Press, 1994.

[14] Varian Corporation, *Diagnosis/Troubleshooting*, Walnut Creek, CA.

[15] Elling, John, Los Alamos National Laboratory, (personal communication).

[16] Robinson, John, Varian Chromatography Systems, Walnut Creek, CA, (personal communication).

[17] Thornberg, Steve, Sandia National Laboratories, (personal communication).

DISTRIBUTION

1	MS	0151	G. Yonas, 9000
1		1002	P.J. Eicker, 9600
1		1003	R.W. Harrigan, 9602
1		1003	R.D. Robinett, 9611
1		1008	D.R. Strip, 9621
1		1010	M.E. Olson, 9622
1		1176	R.D. Palmquist, 9651
1		1004	D.S. Horschel, 9661
1		1006	P. Garcia, 9671
1		1007	A.T. Jones, 9672
1		0501	S.M. Kohler, 2338
5		0501	S.D. Tucker, 2338
5		1010	J.J. Rivera, 9622
10		1003	J. T. Feddema, 9611
1		1003	J.D. Stauffer, 9602
1		0734	J.H. Lee, 6624
1		1436	LDRD Office
1		9018	Central Technical Files, 8523-2
5		0899	Technical Library, 4414
2		0619	Review & Approval Desk, 12630 for DOE/OSTI