CONF-880853--2

UCRL--98069

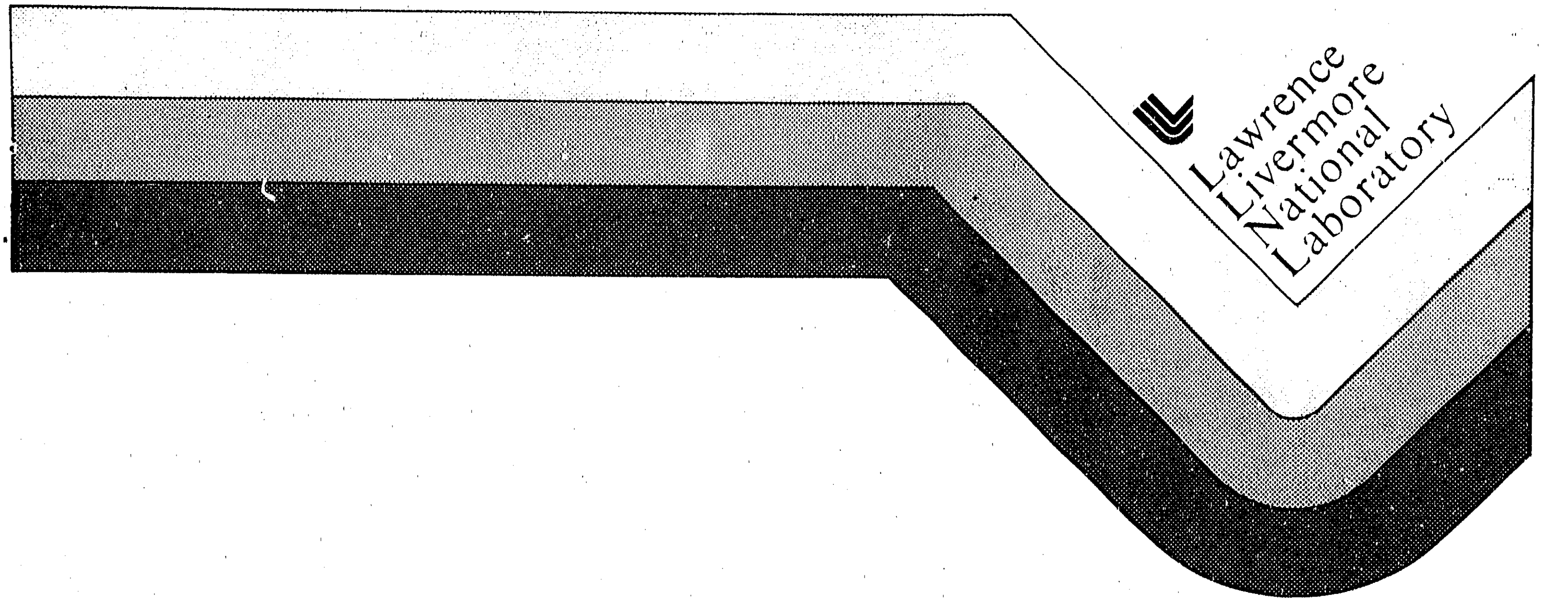
DE91 000817

A SEISMIC EVENT ANALYZER FOR NUCLEAR TEST BAN TREATY VERIFICATION

C. L. MASON
R. R. JOHNSON
R. M. SEARFUS
D. LAGER
T. CANALES

3RD INTERNATIONAL CONFERENCE ON APPLICATION
OF ARTIFICIAL INTELLIGENCE IN ENGINEERING
STANFORD, CALIFORNIA
AUGUST 8 - 11, 1988

JANUARY 1988



A Seismic Event Analyzer for Nuclear Test Ban Treaty Verification

Cindy L. Mason, Rowland R. Johnson
Dept. of Applied Science
University of California, Davis
Livermore, California 94550 USA

Seismic Monitoring Research Program
Lawrence Livermore National Laboratory
Livermore, California 94550 USA

Robert M. Searfus, Darrel Lager, Tom Canales
Seismic Monitoring Research Program
Lawrence Livermore National Laboratory
Livermore, California 94550 USA

Abstract. This paper presents an expert system that interprets seismic data from Norway's regional seismic array, NORESS, for underground nuclear weapons test ban treaty verification. Three important aspects of the expert system are (1) it emulates the problem solving behavior of the human seismic analyst using an Assumption Based Truth Maintenance System, (2) it acts as an assistant to the human analyst by automatically interpreting and presenting events for review, and (3) it enables the analyst to interactively query the system's chain of reasoning and manually perform an interpretation. The general problem of seismic treaty verification is described. The expert system is presented in terms of knowledge representation structures, assumption based reasoning system, user interface elements, and initial performance results.

---

1. INTRODUCTION

The monitoring and interpretation of seismic events is the most reliable means available to verify compliance with treaties regulating underground nuclear weapons testing. However, monitoring for a Comprehensive Test Ban Treaty (CTBT) or Low Yield Test Ban Treaty (LYTBT) increases the requirements on current verification technology.

To evade treaty provisions, nuclear tests would be designed to produce weak seismic signals. These low signal to noise ratio (SNR) events can be hidden in background noise or occur in conjunction with other seismic events. To address this problem sensor technologies offering lower detection levels and improved SNR must be used. However, lowering the detection threshold increases the number of events which must be examined. Our experience indicates up to 20,000 events a year may need to be analyzed from Norway's experimental seismic array, NORESS.

Such events include earthquakes and chemical explosions as well as possible nuclear explosions. Each event must be analyzed to determine if it contained a clandestine nuclear test. There are few experts available for this type of interpretation problem. Therefore, reliable verification of a CTBT or LYTBT will require an automated system for interpreting and classifying seismic events. Such a system, called SEA (Seismic Event Analyzer), is described in the remainder of this paper.

The following section gives an overview of the treaty verification research system at LLNL and motivates our expert systems approach to the problem of seismic interpretation. Sections 4 and 5 examine the problem solving strategy used by seismologists and relate this strategy to belief revision. SEA's knowledge representation and assumption based reasoning schemes are presented in sections 6 and 7. Finally, we present our user interface, some of our implementation experiences, and answer the question, "Does it work?"

2. SYSTEM OVERVIEW

The expert system, SEA, analyzes seismic data from an experimental seismic array called NORESS located in eastern Norway about 100 Km north of Oslo as shown in Figure 1 (Brøding, 1986). SEA presently interprets data from 25 sensors deployed in four concentric rings, the largest being about 3 Km in diameter.

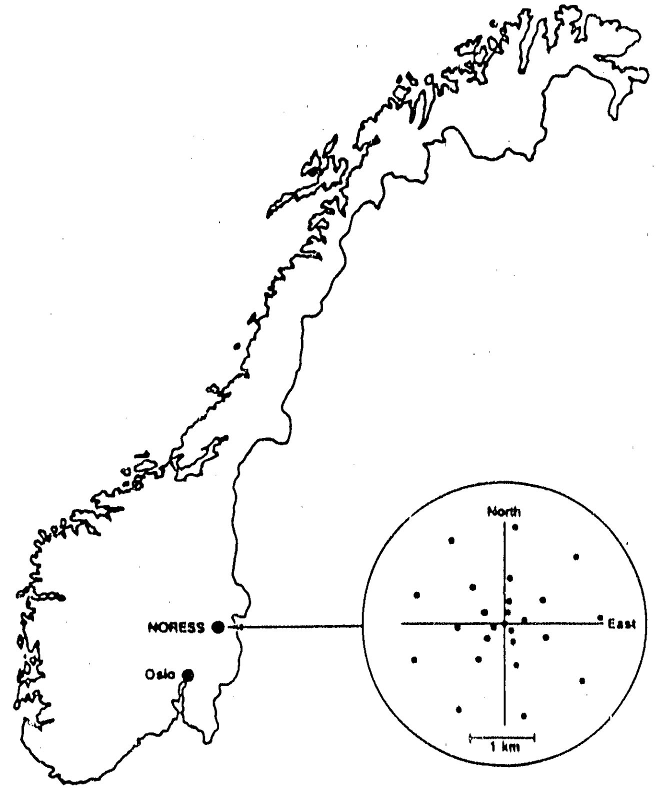
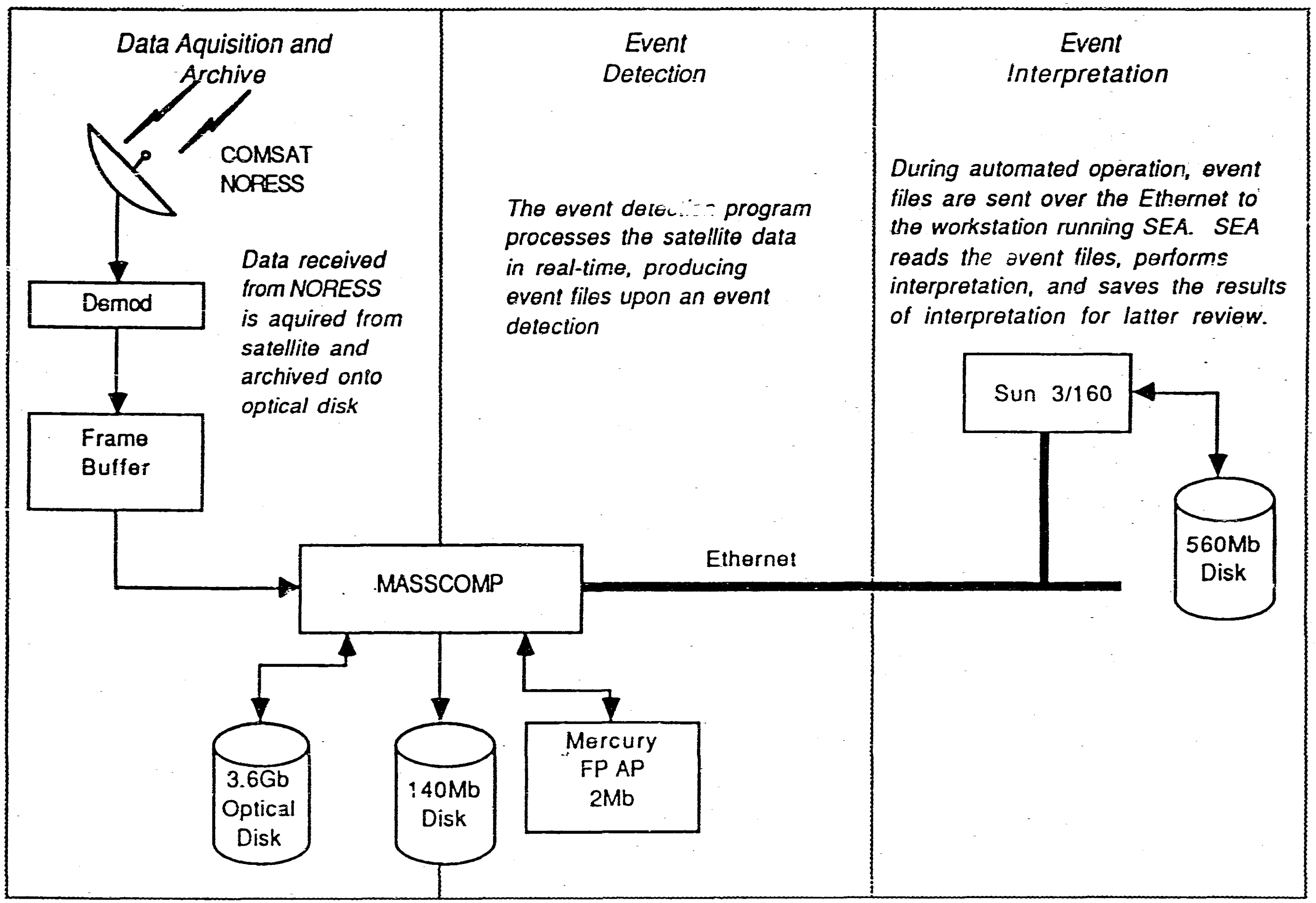


Figure 1. The Norwegian regional seismic array (NORESS). The inset shows the geometry of NORESS with 25 sensors.

Architecture and Data Flow of the Interpretation System



Data received from NORESS is aquired from satellite and archived onto optical disk

The event detection program processes the satellite data in real-time, producing event files upon an event detection

During automated operation, event files are sent over the Ethernet to the workstation running SEA. SEA reads the event files, performs interpretation, and saves the results of interpretation for latter review.

Figure 2: Treaty Verification System Overview

As shown in Figure 2, we receive seismic data from Norway via satellite and archive it onto an optical disk. It is then processed by an event detection program which examines the raw sensor data for a possible seismic event. The event detector program generates an event file for each seismic event detected and sends it via Ethernet to the Sun workstation where the expert system performs the event interpretation. SEA stores the results of the interpretation process in a disk file, giving the seismologist the option to either review the interpretations as they are generated or at a later time.

During the review process, the seismic analyst can examine the chain of reasoning the expert system followed, the waveforms before and after applications of signal processing, and the information comprising an interpretation. The analyst can interactively add new information and direct the expert system to make additional interpretations or modify existing ones. The analyst can also develop an interpretation manually (i.e. without the aid of the expert system) to produce a "second opinion." This philosophy makes maximum use of automation while leaving the final decision to the analyst. SEA's interpretation knowledge was obtained by observing and interviewing seismologists who work with NORESS and is comprised of general seismological knowledge as well as knowledge specific to the NORESS array.

3. A KNOWLEDGE-BASED SOLUTION

3.1 Improved Performance

Previous attempts to automate the task of seismic data analysis using traditional programming techniques rely almost entirely on the algorithmic use of signal processing, for example see Mykkeltveit (1984). Human analysts outperform these programs by using their knowledge about regional propagation characteristics and patterns in the signal or noise to judiciously apply signal processing while forming an interpretation. The human analyst also considers non-geologic information such as the shooting schedule of mines and the breaking up of river ice in the spring. The Knowledge Based system approach enabled us to capture this regional knowledge and drive the signal processing in a pattern driven fashion.

3.2 Incomplete and Uncertain Signal Data

Signal data arriving at a seismic station is characterized at best as incomplete or uncertain. To evade the provisions of a Test Ban Treaty, nuclear tests would be designed to generate weak or possibly altered seismic signals. Depending on the position of the sensors and the path the signals take through the earth, a sensor may not detect all the phases generated by an event. In addition, phases which do arrive may be errorful (intentionally or unintentionally) so that phase measurements and identification are uncertain. These forms of incompleteness and uncertainty are present in non-nuclear events as well. Using a Knowledge Base Systems approach, we are able to treat this error and uncertainty as a natural part of the problem solving process. This approach has been useful in a number of image, speech, and signal interpretation systems, most notably Hearsay-II (Erman, 1980), and is an alternative to more traditional programming techniques where errors result in fatal or exceptional conditions.

3.3 Rapid Prototyping

In our research environment we are concerned with quickly determining the feasibility of the approach while delivering a useful tool. An advantage of the Knowledge Based systems approach is that the architecture supports a knowledge base which will evolve through the life of the system. Initially we developed a proof-of-concept prototype system which has evolved over the past two years into an on-line system which is capable of analyzing real-time satellite data transmissions 24 hours a day.

4. PROBLEM DOMAIN

The goal of seismic data interpretation is to create a hypothesis about the number and location of events which would generate the signals present in a seismogram. To understand how this is done, it helps to visualize what happens during a seismic event.

As a seismic event releases energy, it sets up a number of characteristic waves, or phases, which propagate along various paths through the earth (see Figure 3).

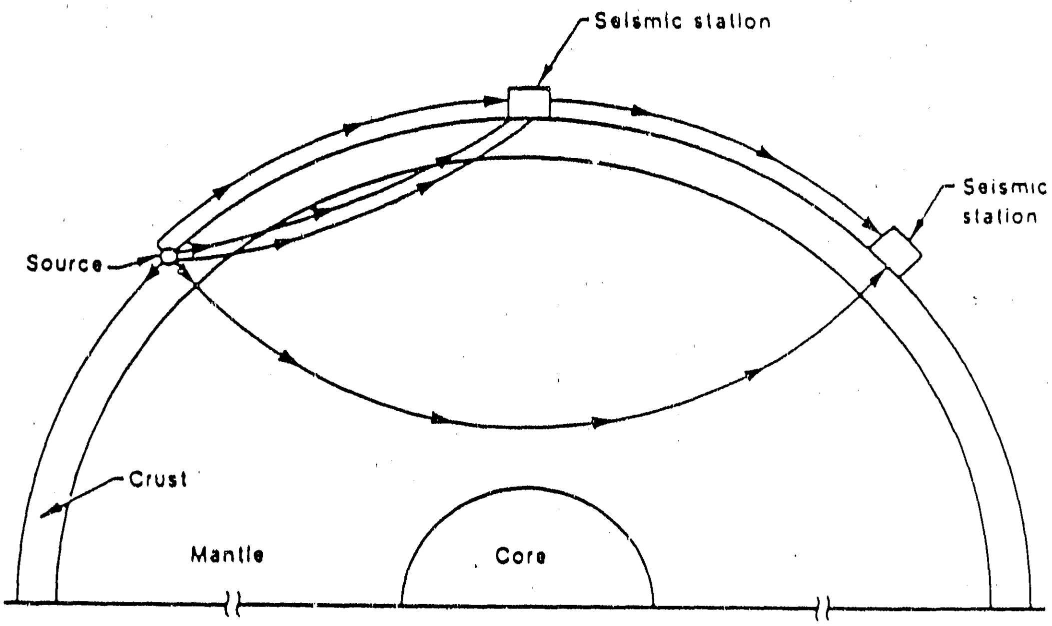


Figure 3. Seismic waves separate and take distinct paths through the earth

Because these waves don't travel the same distance and propagate at the same velocity, they arrive at the seismic sensor at different times. Longitudinal waves (called P for primus) arrive first, followed by transverse waves (S for secundus) and then surface waves (named after their discoverers Rayleigh and Love)(see Figure 4). It is this difference in arrival time which enables a seismologist to estimate the distance of an event. The further apart the phases are, the farther they had to travel to reach the station. Direction is determined using a number of phase characteristics calculated with signal processing tools. The location of a seismic source is then estimated using distance and direction.

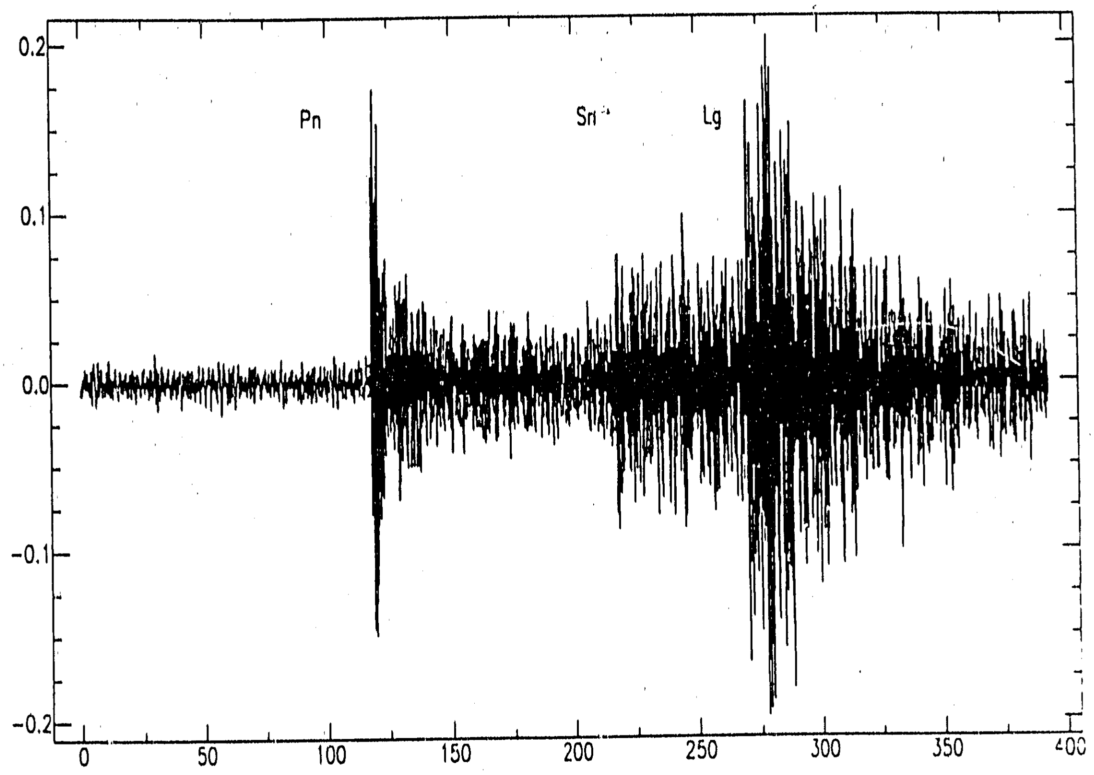


Figure 4. Seismogram depicting three distinct wave arrivals.

However, formation of an interpretation is further complicated when signals from multiple overlapping events arrive at the station. Not all phases from the events will arrive, and those that do arrive can be of poor quality. In order to form an interpretation, the seismologist needs to identify phase type and associate phases as belonging to the same event. Furthermore, the events that are of most interest have a low signal to noise ratio. In essence, a seismologist finds an interpretation by hypothesizing and testing interpretations. Figure 5 shows the overall process used by the seismologist in analyzing a seismic event. In general, the process is one of sequentially building a series of partial interpretations, each an extension of one before it. Each partial interpretation is checked for consistency and discarded if shown to be inconsistent.

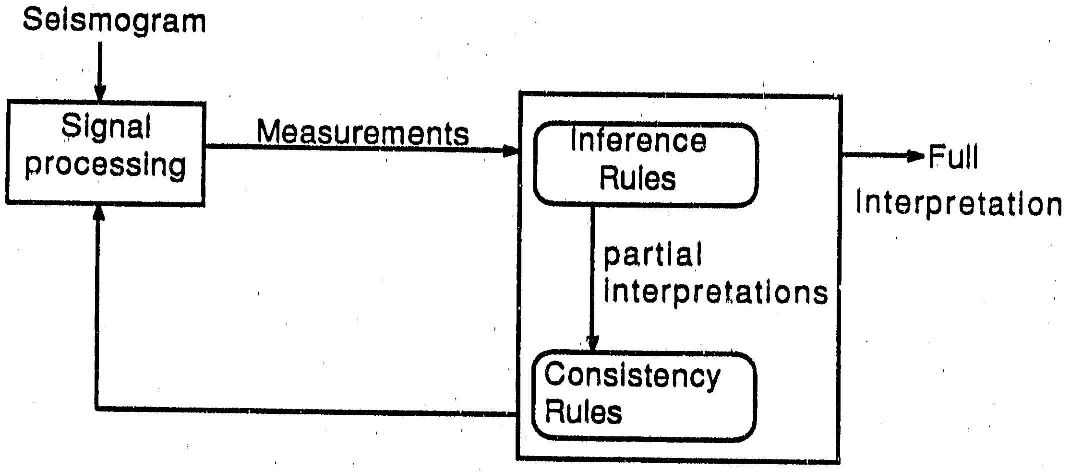


Figure 5. General interpretation process used by seismologist in analyzing a seismic event

Initially a core set of signal processing routines are applied to the seismic data. This yields information that is used to make an initial set of partial interpretations. These form the basis for additional signal processing to be done that will be used to extend the partial interpretation. This cycle continues until a partial interpretation is disproved or extended to a full interpretation. A full interpretation that has not been disproved constitues a valid interpretation of the seismic data.

5. TRUTH MAINTENANCE SYSTEM

The process whereby a seismologist hypothesizes a partial interpretation and determines its validity can be formally described by a belief revision system. A partial interpretation is disproved if the belief in its validity leads to a contradiction. For example, it is known that a Pn phase must have a velocity of greater than 6 Km/sec. Suppose that it is believed that a phase is Pn in a partial interpretation and that gives rise to a inference chain resulting in the fact that the phase has a velocity less that 6 Km/sec. Then it must be the case that the belief is incorrect and the partial interpretation is discarded. Truth maintenance systems (TMS) [DOYL79] have been shown to handle this kind of belief revision by maintaining records about dependencies between facts and detecting contradictions among believed facts. SEA uses a particular form of TMS called an assumption based TMS (ATMS) [DEKL84].

Typically, ATMSs are most appropriate in situations where there are a number of solutions and all must be found. In situations where there is only one solution or where only one must be found an ATMS is not as efficient as a TMS[DEKL84]. In theory, there is a single correct interpretation of a given seismic event. However, in practice multiple interpretations are unavoidable. In fact, seismologists are unable to arrive at a single interpretation when there is insufficient information to exclude invalid interpretations. Multiple valid interpretations from SEA indicate that there are not enough rules to reject invalid interpretations.

6. KNOWLEDGE REPRESENTATION SCHEME

6.1 SEMANTIC NETWORK

A semantic network formalism is used as the representation scheme for a seismic interpretation. As shown in Figure 6, links are directed and go from a single node to a single node. There are two types of nodes; atomic nodes and non-atomic nodes. Non-atomic nodes represent seismological concepts such as EVENT, PHASE, or SEGMENT. Atomic nodes are used to represent a value, such as begin-time or number-of-phases, and only have links pointing into them.

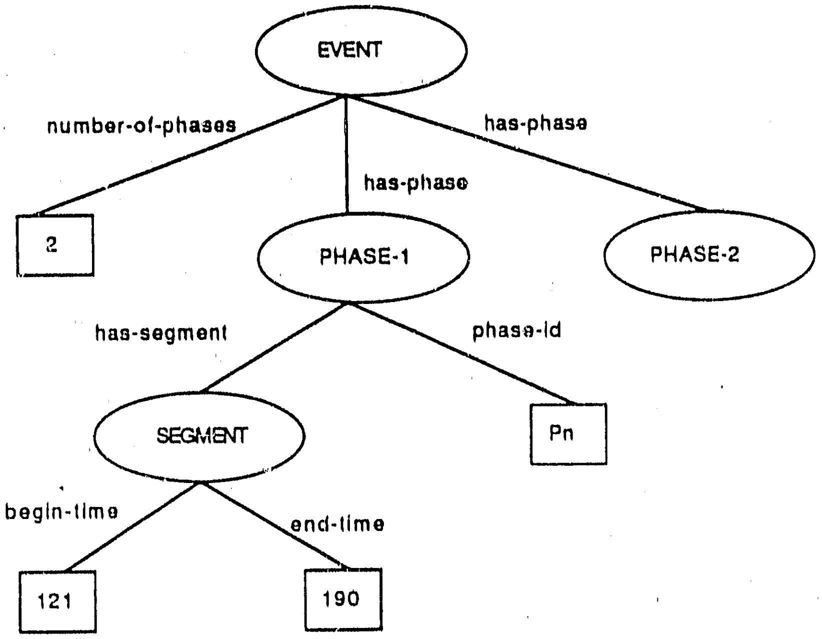


Figure 6. The Semantic Network indicates the EVENT has 2 PHASES. PHASE-1 is shown with its begin and end times.

For the purpose of accessing and processing the network in a rule, we define a fact to be a <node, link, node > triple. A network is completely specified by the set of facts that it contains. Thus the net in Figure 6 consists of the facts

```
<EVENT,   number-of-phases, 2>
<EVENT,   has-phase,        PHASE-1>
<EVENT,   has-phase,        PHASE-2>
<PHASE-1, has-segment,      SEGMENT>
<PHASE-1, phase-id,         Pn>
<SEGMENT, begin-time,       121>
<SEGMENT, end-time,         190>
```

6.2 IF-THEN RULES

IF-THEN rules are the knowledge representation scheme used to encode the interpretation knowledge. Each rule has the form

(*rule-name pattern action*)

During program execution, the *pattern* portion of rules are matched against the nodes and links of the semantic network to determine which rules will have their *action* portion executed. Consider a rule to compute the velocity of a Pn phase using the semantic network in Figure 6.

```
(compute-velocity-for-Pn
      ((EVENT has-phase PHASE-1)
       (PHASE-1 phase-id Pn))
      (ASSERT (PHASE-1 velocity (compute-vel-using-beamform)))
```

The *pattern* portion of the rule matches the facts describing an EVENT which has a PHASE whose phase-id is Pn. The *action* is to invoke a function (which may be written in LISP, C or FORTRAN) to compute the phase's velocity using a signal processing technique, beamforming. The computed value is then added into the network as the fact <PHASE-1, velocity, value> via the "ASSERT" function.

7. ASSUMPTION BASED REASONER

The assumption based reasoning techniques of SEA are a hybrid of the concepts found in [STALL71] and [DEKL84]. The system tags each deduced fact with a justification indicating the rule and antecedent facts used to create it and the assumptions under which the fact is believed. This information is useful in providing an explanation facility as well as a trace of rule operation for debugging and understanding the system's behavior. This same information is also used to determine the current set of beliefs and in implementing dependency-directed backtracking.

The ATMS is composed of an assumption database and TMS rules. Control of rule firings is achieved in a forward chaining manner, using antecedent demons to monitor the facts data base and instantiation queues to control the order of rule firings.

7.2 The Assumptions Data Base

The Assumptions Data Base contains an assumption set for each fact inferred into the Semantic Network. The assumption set for a fact indicates the assumptions upon which its belief ultimately depends (i.e. a fact is believed if its assumption set is believed). More formally, an assumption set is defined to be a subset of the known facts

$$F = \{f_1, f_2, ..., f_n\}$$

Thus, the set of assumption sets is the power set of F. An assumption set A is labelled either BELIEVED or DISBELIEVED. BELIEVED means that facts of A are consistent with one another. i.e. they can all be believed at the same time. DISBELIEVED means that the facts of A are inconsistent, i.e. they can not all be believed at the same time. Let A and B be assumption sets with $A \subset B$. If A is DISBELIEVED then B must also be DISBELIEVED since the subset of facts of A that cause it to be DISBELIEVED will also be in B.

Each fact f has an assumption set A(f) associated with it. In essence, f is believed if A(f) is BELIEVED and is disbelieved if A(f) is DISBELIEVED. The manner in which assumption sets are assigned to facts, their labels and possible subsequent re-labelling provides the mechanism by which assumption based reasoning or assumption based truth maintenance is realized. When a rule fires there is the set of antecedent facts $a_1, ..., a_n$ and the set of created facts $c_1, ..., c_m$. Each c_i can be created by one of three actions each producing a different value for $A(c_i)$. The actions and the values for $A(c_i)$ are:

$$ASSERT(c_i)$$

$$A(c_i) = \cup_{j=1}^{n} A(a_j)$$

$$ASSUME(c_i)$$

$$A(c_i) = \cup_{j=1}^{n} A(a_j) + c$$

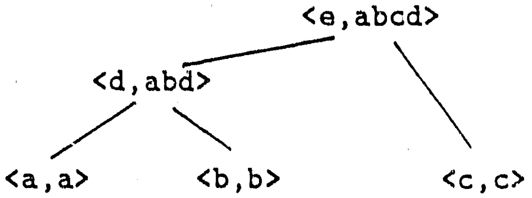
$$GIVEN(c_i)$$

$$A(c_i) = c$$

For example, consider the following inferences

$$\begin{aligned} \perp &\rightarrow GIVEN(a), GIVEN(b), GIVEN(c) \\ a \wedge b &\rightarrow ASSUME(d) \\ d \wedge c &\rightarrow ASSERT(e) \end{aligned}$$

This gives rise to $< fact, assumptionset >$ pairs



7.3 Truth Maintenance Rules

The right hand side actions thus far have been concerned with creating facts. The system is monotonic in that facts are never deleted. Non-monotonic reasoning is achieved by changing the label value of an assumption set. This is accomplished through a special class of rules called TMS rules. TMS rules take the same form as inference rules. In a TMS rule, pattern specifies a group of believed facts which constitute an inconsistency. If the pattern successfully matches against the semantic network, an inconsistency among facts has occurred and the action CONTRADICT is invoked to repair the Assumptions Data Base.

For example, if the instantiation

$$a_1, a_2, ..., a_n \rightarrow CONTRADICT$$

fires, then the assumption set $\cup_{j=1}^{n} A(a_j)$ is marked DISBELIEVED. In the above example, suppose we now have

$$b \wedge c \rightarrow CONTRADICT$$

The assumption sets $\{b, c\}$ as well as $\{a, b, c, d\}$ are marked DISBELIEVED. Thus, the fact e that was BELIEVED is now DISBELIEVED.

To illustrate in terms of the seismic domain, suppose after computing the velocity for the phase in the "compute-velocity-for-Pn" rule from section 6.2 we realize the value contradicts the belief that the phase-id is Pn. The following truth maintenance rule encodes the inconsistency between the two facts.

```
(Pn-velocity-consistency
          (Believed (EVENT has-phase PHASE-1)
                    (PHASE-1 phase-id Pn)
                    (PHASE-1 velocity < 6.5))
          (CONTRADICT))
```

At this point, the assumption sets for the facts involved are no longer believed. Any other facts dependent upon these assumptions become disbelieved as well. Initially, all facts are *Unknown*, meaning they do not exist in the database yet and are neither *Believed* or *Disbelieved*. A created fact will have the status *Believed* until its assumption set becomes involved in a contradiction.

7.4 Queue-oriented Control

Control of rule firing is achieved using antecedent demons to monitor the facts data base and instantiation queues for inference and truth maintenance rules. Rules are implemented as instantiation frames whose slots are variables waiting to be bound. Initially instantiations are unsatisfied and reside in the "unsatisfied" queue. Each time a fact is added to the data base, the antecedent demon tries to satisfy instantiation slots using the fact as a variable binding. Once all slots are finally satisfied, the rule is triggered and can do one of two useful things: detect a contradiction or create new facts.

Contradiction rule instantiations move to the "TMS-ready-to-fire" queue where they will all be executed in the current inference cycle. Inference rule instantiations on the other hand move to the "INF-ready-to-fire" queue and may not fire immediately , since they are subject to prioritization. Inference instantiations are inserted into the queue in the order in which they are to fire, the top-most instantiation being the next to fire.

Once a rule has fired, the instantiation moves the "already-fired" queue. Hence the queues provide a complete memory of how rules were applied in solving a particular problem instance.

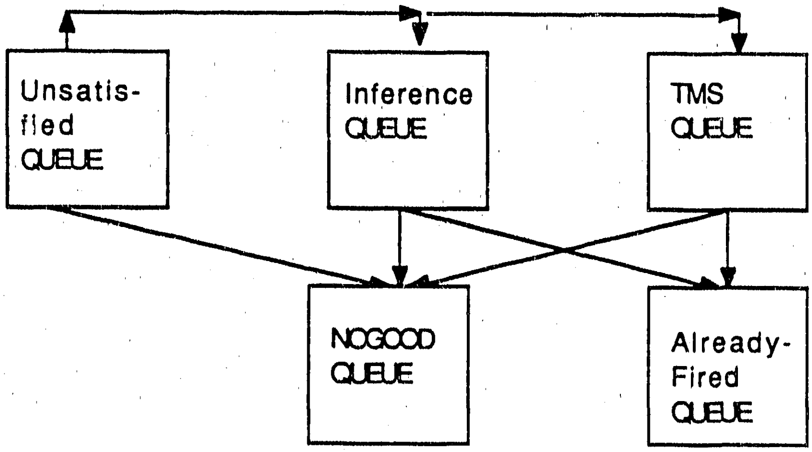


Figure 7 Instantiations frames migrate among queues during program execution.

7.5 Belief Revision During Pattern Matching

During the time an instantiation is waiting to be complete satisfied, the belief status of the previously bound variables may change. Hence after each submatch $\cup_{i=1}^{k} A(a_i)$ is checked, where k is the number of submatches so far. If it is DISBELIEVED then $\cup_{i=1}^{k} A(a_i)$ will be DISBELIEVED and the instantiation may be discarded and moved to the "NOGOOD-instantiation" queue. This prevents any further pattern matching on partially instantiated rules that can never fire. A similar circumstance can occur while a satisfied inference rule is waiting to be fired in the "INF-ready-to-fire" queue.

This inferencing technique will find all consistent interpretations and is independent of the order in which instantiations fire. However, the order has a profound effect on the amount of time required to find these interpretations. Recall that if assumption sets A and B are such that $A \subset B$ and A is DISBELIEVED then B is DISBELIEVED. Suppose $a_1, ..., a_n - > c$ where c is ASSERTed or ASSUMEd and $\cup_{i=1}^{n} A(a_i)$ is DISBELIEVED. Then A(c) must also be DISBELIEVED. The components of c may require expensive computations that are ultimately wasted since c is DISBELIEVED. This useless work is eliminated by checking $\cup_{i=1}^{n} A(a_i)$ before the action of the instantiation is evaluated. If it is DISBELIEVED the instantiation is discarded and c is never created.

7.6 Observations

In general, there is more than one interpretation generated for each event. This introduces the problem of assigning a confidence measurement of interpretations derived by the system. The usual approach taken in expert systems is to provide confidence values to facts that are in turn derived from the rule and confidence values of antecedent facts [SHORT76]. In an assumption based system a single fact may have multiple combinations of rules and antecedent facts each inferring the fact or its negation. Therefore, the credibility of a single fact will depend on all rules and antecedent fact credibilities. The nature of this dependability has not been addressed, although the analysis found in [GINS84] may be applicable.

SEA uses a different approach in determining interpretation credibilities. In essence, the strategy in evaluating an interpretation is to make all possible inferences about an interpretation trying to find a reason to discard it. Although rules may vary in discrimination power, in general the more inferences made without encountering a contradiction the more credible the interpretation becomes. Therefore, the total number of inferences made on an interpretation is a measure of the interpretation's credibility.

A single contradiction is enough to cause work on an interpretation to halt. It is often the case, however, that there are several contradictions that can be generated for an interpretation. Due to the assumption based architecture it is possible to find some or all of these contradictions. A single fact can have several DISBELIEVED assumption sets. Even after inferencing has stopped for an interpretation assumption sets that support facts already in the interpretation can be marked DISBELIEVED. The total number of contradictions associated with an interpretation provides a measure of an interpretation's incredibility.

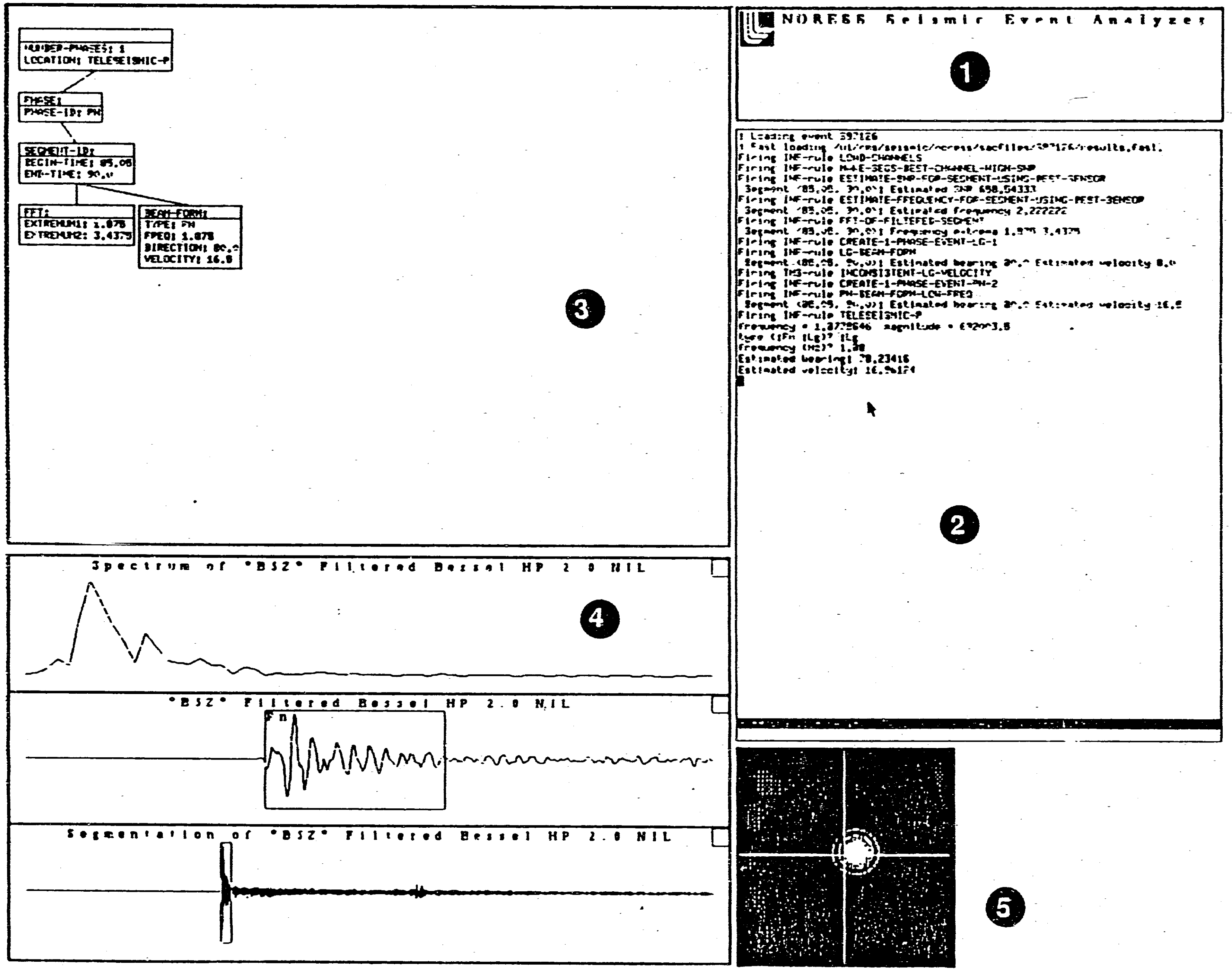
8. USER INTERFACE

SEA is designed to operate in two fundamental modes: automatic and interactive. When started in automatic mode SEA processes events in real-time without human interaction. From interactive mode, SEA can be used as an assistant for manually processing seismic data, or may be used by an analyst to review results processed during automated operation.

The analyst reviews events processed in automated operation by running SEA in its interactive mode. In this mode, the display screen is divided into five windows (shown in Figure 8):

1. Logo Window - The Lawrence Livermore National Laboratory's logo and the Expert System's title are displayed in this window.

Figure 8. SEA's User Interface



```
! Loading event 397126
! Fast loading /ut/rms/seismic/noress/sacfiles/397126/results.fast.
Firing INF-rule LOAD-CHANNELS
Firing INF-rule MAKE-SEGS-BEST-CHANNEL-HIGH-SNR
Firing INF-rule ESTIMATE-SNR-FOR-SEGMENT-USING-BEST-SENSOR
 Segment (85.05. 90.0): Estimated SNR 658.54333
Firing INF-rule ESTIMATE-FREQUENCY-FOR-SEGMENT-USING-BEST-SENSOR
 Segment (85.05. 90.0): Estimated frequency 2.222222
Firing INF-rule FFT-OF-FILTERED-SEGMENT
 Segment (85.05. 90.0): Frequency extrema 1.875 3.4375
Firing INF-rule CREATE-1-PHASE-EVENT-LG-1
Firing INF-rule LG-BEAM-FORM
 Segment (85.05. 90.0): Estimated bearing 80.0 Estimated velocity 8.0
Firing INS-rule INCONSISTENT-LG-VELOCITY
Firing INF-rule CREATE-1-PHASE-EVENT-PN-2
Firing INF-rule PN-BEAM-FORM-LOW-FREQ
 Segment (85.05. 90.0): Estimated bearing 80.0 Estimated velocity 16.8
Firing INF-rule TELESEISMIC-P
frequency = 1.8729646  magnitude = 692063.8
type (:Pn :Lg)? :Lg
frequency (Hz)? 1.88
Estimated bearing: 78.23416
Estimated velocity: 16.96174
```

2. Editor Window - The editor Emacs is used to run and communicate with SEA. Progress of event interpretation, such as names of the rules that execute, high-level results of signal processing, and debugging information, is displayed in this window. Rules and system code may be edited and debugged in Emacs without having to exit SEA.

3. Semantic Network Display Window - Once an event interpretation has been made, information from the semantic network associated with that event is graphically displayed in this window. The displayed information is organized in a hierarchical fashion reflecting its structure in the semantic network.

4. Waveform Display Stack - Three windows, stacked upon each other, are used to display results of signal processing during interpretation. A waveform is displayed by scrolling the top two windows down, and plotting the new waveform in the top window. This stacking mechanism provides a historical overview of signal processing activity during event interpretation.

5. FK Analysis Contour Display Window - This window displays a contoured image of the surface resulting from frequency-wavenumber (FK) analysis. This analysis technique is used to determine the direction and velocity of a seismic phase and the image displayed in this window provides feedback on the quality of the direction/velocity estimate.

The analyst's input to SEA's interactive interface is performed primarily by pointing and selecting with the workstation's mouse. Waveforms may be examined and selected with the mouse for analysis, information may be obtained about items in the semantic network by "clicking" the mouse in specific regions of the semantic network display, and several actions can be invoked through the use of menus.

At the beginning of an interactive session, SEA presents a menu of events that are available for review. When an event is selected from the menu, SEA reads the event's data with any results saved from automated processing. SEA processes the event in a manner identical to automatic mode, but runs faster when results are available from a previous analysis.

Once processing completes, a menu of interpretations is displayed with the word "CONSISTENT" beside an interpretation if it is believed, or the word "INCONSISTENT" if it is disbelieved. Selecting an interpretation from the menu displays the event waveform with superimposed phase information on the waveform display stack, and associated information from the semantic network is displayed in the semantic network display window. Furthermore, if the selected interpretation is disbelieved, the inconsistencies that caused the interpretation to be disbelieved are also shown in the semantic network display window.

9. DOES IT WORK?

We have begun to perform extensive testing to determine if SEA will acceptably perform the task it was designed for and identify areas of further improvement in the processing strategy, knowledge representation, and user interface. Initial results obtained from testing during development and operating in real-time follow.

9.1 Developmental Testing

Archived data was used during the development and refinement of SEA's rule set to determine how close the system followed the expert's reasoning in estimating event locations, and how efficiently the system used time expensive computations in the interpretation process. Several events from the archived data exhibiting a variety of signal conditions and source locations form the basis of a test set used to determine how changes in the rules effect the overall interpretation performance.

This off-line processing has enabled us to correct and tune the interpretation strategy employed by SEA. Our experience now shows that the system has evolved enought to analyze events and estimate source locations with a high degree of reliability. Some events from the test data set are shown in Figure 9, and illustrate the variety of events the system must be able to interpret.

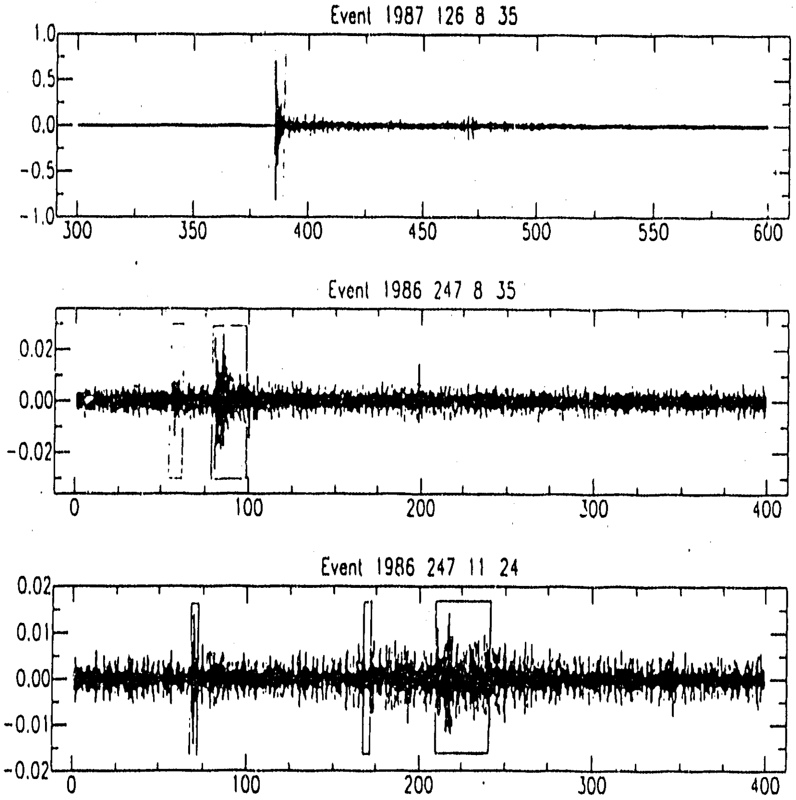


Figure 9. Events from the test data set include a Teleseismic P (top), an event from Southern Norway (middle), and an event from a mining explosion in the Leningrad region of the USSR (bottom). Events are labeled with the Greenwich Mean Time (year, Julian day, hour, and minute) associated with the beginning of the data file.

The first event (1987 126 8 35) is from an announced Soviet underground nuclear test at the USSR Semipalatinsk Test Site. The interpretation provided by SEA characterized the event as a teleseismic P phase (indicating a source distance near or greater than 2000 Km) with an estimated velocity of 16.5 Km/s and a bearing of 80 degrees Northeast. Our domain expert interpreted the event using a different set of analysis tools and obtained a similar characterization with a velocity of 16.8 Km/s. Analysis of the second event (1986 247 8 35) by SEA characterized the event as a Pn/Lg combination, and estimated the location of the source at latitude 59.3 degrees north, longitude 9.6 degrees east (in Southern Norway). The location provided by the expert was 59.2 degrees north, 9.4 degrees east. The final event (1986 247 11 24) was interpreted by the expert system as a Pn/Sn/Lg combination located at 61.1 degrees north, and 29.0 degrees east (in the Leningrad region of the USSR). The expert's interpretation concurs with SEA's, locating the event at 60.0 degrees north, and 29.6 degrees east.

9.2 Real-time Testing

To determine how SEA would perform in an operational environment, SEA was run nearly continuous for five days in on-line automated mode. While the automated system was interpreting real-time data, SEA's interactive facilities were used to review the automated mode's interpretations and arrive at final conclusions for each detection processed. During the evaluation period, the event detection process was triggered eighty four times by the real-time data stream and sent the event detection to SEA for interpretation. A summary of the results of this evaluation is presented in Table 1.

Total events detected	84
False alarms	31
Events missed	2
Unacceptable interpretations	5
Acceptable interpretations	46

Table 1. Summary of real-time evaluation.

Over one third of the detections were attributed to false alarms which are typically caused by abnormalities such as errors introduced in data transmission, or are a result of surface waves produced by a seismic source local to the array such as a car or train. SEA has demonstrated the ability to reject most detections caused by data abnormalities. False alarms caused by local sources usually lack the phases necessary to locate the source, but when detected by SEA, are interpreted as single phase events. Since the number of false alarms increases significantly as the detection threshold is decreased, the proper handling of false alarms is important in prioritizing and filtering interpretations presented for review.

SEA clearly missed two events, both teleseismic P phases with low signal to noise ratios. We have since modified the processing that the expert system uses to detect such phases, and are encouraged by the initial results. Five interpretations were not acceptable without refinement by the expert, but in each case the interpretation that was provided was useful in drawing the final conclusions.

The remaining forty six interpretations SEA produced in automated mode were considered acceptable by our domain expert. Distances of the events processed range from 50 to over 1000 Km and a large number of directions were covered. Those events that were not teleseismic are shown by location in Figure 10. Areas with a large numbers of detections (indicated by clusters of *'s) suggest frequent seismic activity such as that produced by mining or construction.

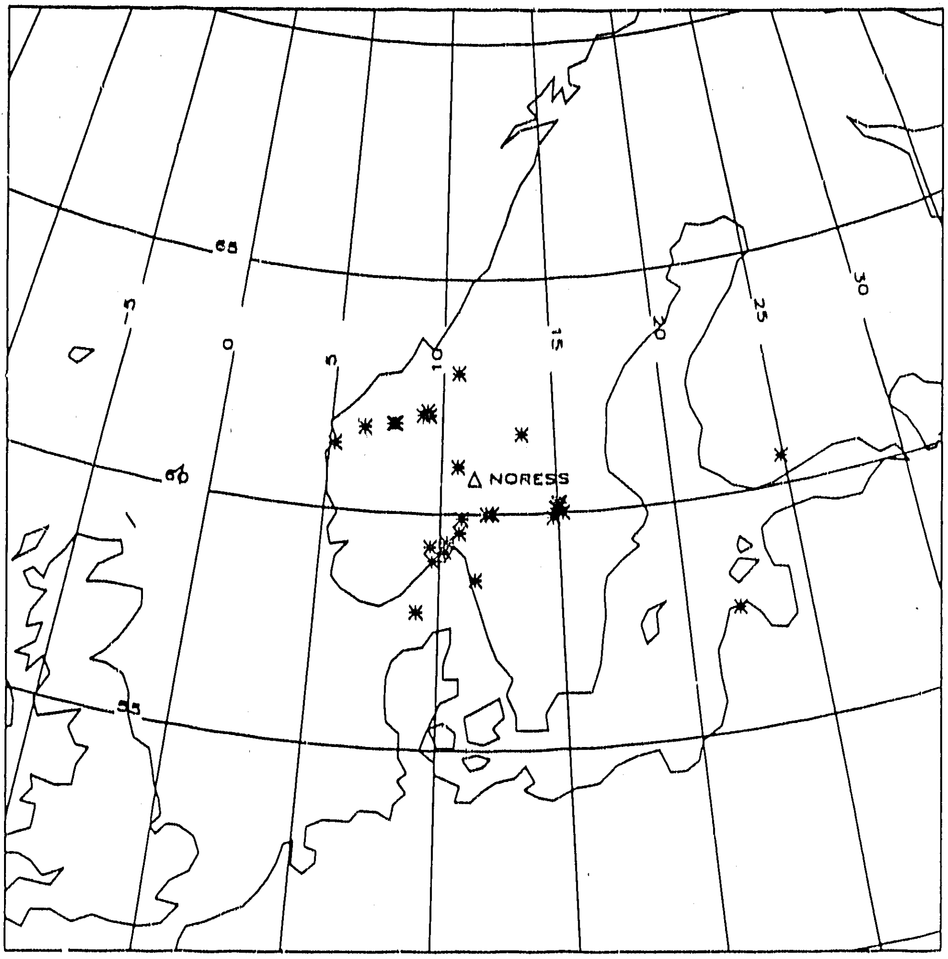


Figure 10. Event locations provided by SEA's automated mode.

Results observed from off-line developmental testing and real-time operation are very encouraging, and tend to indicate that SEA does an acceptable job interpreting and locating seismic events using the NORESS array. More evaluation is planned for SEA, including a comparison of results obtained from other agencies using the NORESS array.

10. Implementation

SEA was originally prototyped on a Symbolics 3600 LISP Machine using ZETALISP, Flavors, and the Symbolics LISP Development Environment. The system was ported to the Sun Microsystems family of workstations under Common LISP for delivery. The Sun UNIX environment is based on the MIT X Window System, and the GNU Emacs editor. Several parts of the original system were rewritten in C or FORTRAN, and are accessed from LISP through a foreign function interface. The expert system requires approximately twenty to twenty five megabytes of virtual memory, and performs well with eight to sixteen megabytes of physical memory. SEA requires a floating point coprocessor for signal processing, and has support for the Sun floating point accelerator.

The major components of SEA are shown in Table 2 below.

Module	Implementation Language
ATMS Inference Engine	LISP
Signal and Image Proc.	C, FORTRAN, LISP
Graphics	C, LISP
File I/O, Network I/O	C, CSH
Seismic Specific Support	LISP
Site Specific Support	LISP
Rules	LISP Based Production Rule Langauge

Table 2. SEA Component Modules and the Implementation Languages

11. SUMMARY

We have presented the SEA expert system as an intelligent assistant for the task of seismogram analysis for test ban treaty verification. The program architecture of the expert system enabled us to mimic the problem solving behavior of the seismologist when analyzing the NORESS data. While the system is still undergoing further testing and development, our preliminary results indicate expert systems are a viable technology to be used in automatically analyzing the enormous amount of data necessary for seismic verification. We are also currently developing a distributed Assumption-based Truth Maintenance System for using a group of the expert systems to interpret data from a network of seismic stations (Mason, 1986).

Acknowledgements. This work is a result of the contributions of many people. In particular we wish to thank our expert seismologists, Terri Hauk and Fred Followill, and our signal processing experts, Dave Harris and Farid Dowla. In addition we would like to thank Jay Zucca for helpful comments and suggestions in the writing, Chuck Grant for TEX advice, and Betsy Hickman and Terri Hauk for graphics support. Last but certainly not least, we wish to thank the people who funded this project, the Department of Energy for the Treaty Verification Program, and the Regents of the University of California for graduate research support. This work has been performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

References

Breding, Dale R. (1986). First 1.5 Years With The Norwegian Regional Seismic Array, *Proceedings 1986 RSTN/NORESS Research Symposium.*

deKleer,J. (1986). An Assumption Based TMS, *Artificial Intelligence,* 28, No. 2.

Doyle,J. (1979). A Truth Maintenance System, *Artificial Intelligence,* 12, No. 3.

Erman, L.D., Hayes-Roth, F., Lesser, V.,R., and Reddy, R. (1980). The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty, *Computing Surveys,* 12:213-253, June.

Ginsberg, Mathew, (1984). Non-Monotonic Reasoning Using Demptster's Rule, *Proceeding of the National Conference on Artificial Intelligence,* Austin, Texas, August.

Mason, C. (1986). Cooperating Intelligent Agents - An Overview and Proposal for Research, *Technical Report UCID-20928,* U.C. Davis, Dept. of Applied Science, Livermore, CALIF. 94550.

Mykkeltveit, S., and Bungum, H., Processing of Regional Seismic Events Using Data from Small-Aperture Arrays, *Bulletin of the Seismological Society of America,* 74:2313-2333.

Shortliffe, E. H.(1976). *Computer Based Medical Consultation:MYCIN,* New York: American Elsevier, 1976.

Stallman, R., and Sussman, G. (1971). Forward Reasoning and Dependency-Directed-Backtracking In a System For Computer Aided Circuit Analysis, *Artificial Intelligence,* 9, No. 2.

END

DATE FILMED

11 / 06 / 90