AI Magazine Volume 3 Number 2 (1982) (© AAAI)

Signal-to-Symbol Transformation: HASP/SIAP Case Study

H. Penny Nii
Edward A. Feigenbaum
Department of Computer Science
Stanford University
Stanford, California 94305

John J. Anton
A. J. Rockmore
Systems Control Technology, Inc.
1801 Page Mill Road
Palo Alto, California 94303

ARTIFICIAL INTELLIGENCE is that part of Computer Science that concerns itself with the concepts and methods of symbolic inference and symbolic representation of knowledge. Its point of departure—its most fundamental concept—is what Newell and Simon called (in their Turing Award Lecture) "the physical symbol system" (Newell and Simon, 1976).

But within the last fifteen years, it has concerned itself also with signals—with the interpretation or understanding of signal data. AI researchers have discussed "signal-to-symbol transformations," and their programs have shown how appropriate use of symbolic manipulations can be of great use in making signal processing more effective and efficient. Indeed, the programs for signal understanding have been fruitful, powerful, and among the most widely recognized of AI's achievements.

Many different people helped in building HASP/SIAP in many different capacities. The people acknowledged below are project leaders (*), consultants, and programmers who waded through the myriad of technical problems and codes We are also indebted to John Miller, Jay Seward, Dan Sestak, and Ken McCoy—our experts They cheerfully took on the frustrating and time-consuming jobs of having their brains picked by knowledge engineers, and their own performance then outstripped by the program Those who helped with HASP include John Anton, Scottie Brooks, Edward Feigenbaum, Gregory Gibbons, Marsha Jo Hanna, Neil Miller, Mitchell Model, Penny Nii*, and Joe Rockmore Those who helped with SIAP include John Anton*, Al Bien, Scottie Brooks, Robert Drazovich*, Scott Foster, Cordell Green, Bruce Lowerre, Neil Miller*, Mitchell Model, Roland Payne, Joe Rockmore, and Reid Smith

HASP[1], and its follow-on, SIAP, are among these programs. HASP arose from an important national defense need. It appeared to be impossible to satisfy the computational requirements of a major ocean surveillance system of sensors (at least within the bounds of economic feasibility) with conventional methods of statistical signal processing. AI's signal understanding methods were maturing in the early 1970s. Vision research had been underway for several years. The ARPA Speech Understanding Project was well into its first phase (Newell et al., 1973). And the DENDRAL project for the interpretation of mass spectral data in terms of organic molecular structures had achieved significant success in certain narrow areas of chemical analysis (Lindsay, Buchanan, Feigenbaum, and Lederberg, 1980). The time was ripe to attempt the application of the emerging techniques to the ocean surveillance signal understanding problem. This insight was made by Dr. Lawrence Roberts, then Director of Information Processing Techniques for ARPA

At his request, and with ARPA support, scientists at the Stanford Heuristic Programming Project, with the help of scientists at Systems Control Technology, Inc. (SCI), began in 1972 to study the feasibility of the project. System design and programming began in 1973. The project was located at

[1] In earlier literature, HASP was referred to as SU/X (Feigenbaum, 1977; Feigenbaum, 1980; Nii and Feigenbaum, 1978)

SCI because of the company's expertise in the military problem and because of the classified nature of the work. Feigenbaum was the principal investigator, and Nii was responsible for the detailed design and much of the programming. Scottie Brooks also contributed significant programming. The primary expert in this Expert System project was John Miller, a recently retired officer from the military. Scottie Brooks acquired expertise about acoustic and other data charateristics and took over some of the expert's role during SIAP development. Many others (mentioned above) contributed.

The first year was spent in understanding the nature of the signals, the signal-generating objects, the symbolic context in which the signal analysis was taking place, and in demonstrating that one could not "DENDRALize" this problem. Systematic generation and pruning of the hypothesis space was not the appropriate model. But we learned a great deal, and were able by the end of the year to recognize the appropriate framework when it presented itself. That framework was the "blackboard" model of the HEARSAY-II effort in speech understanding being done at Carnegie-Mellon University (CMU) (Erman, Hayes-Roth, Lesser, and Reddy, 1980; Lesser and Erman, 1977).

The second year and beyond consisted of a rush of activity to program, adapt, and alter the CMU model to fit the problem at hand, to finish the acquisition and encoding of the knowledge, and to perform a series of tests to demonstrate and validate the work. The HASP phase ended in the fall of 1975.

SCI scientists continued the work in SIAP, which began in 1976. The HASP project had intentionally evaded one difficult part of the overall signal-to-symbol transformation problem, the part that is sometimes called "low-level" processing, the processing activity closest to the signal data. HASP never saw real signals. It saw descriptions of signals, albeit "low-level" descriptions. The identification of line segments and their characterization were done by people. SIAP was an attempt to automate this phase, and involved as well some necessary modifications to HASP to allow it to cope with this additional confrontation with reality. The SIAP work was complicated by the fact that the SCI scientists were constrained by their sponsor to use signal processing programs that had been developed in another context by another ARPA contractor The SIAP effort ended early in 1980 after showing significant demonstration on real ocean data in real time.[2]

[2]Some purists insist that the only valid use of the term "signal" is the set of digitized voltages that arise from the normal functioning of the sensor. But we are not that rigid in our view We are willing to consider as "signal" those low-level algorithmic transformations of the "raw data" that are in the standard tool-kit of the signal processing community (sush as the Fast Fourier Transform). Indeed, the humans who do the HASP/SIAP task manually never see "the numbers" either Their job begins after some elementary transforms have been computed This is not to imply that there isn't additional leverage in pushing AI methods one last level closer to "the numbers." But to assert that HASP/SIAP is not really looking at "the signal" is to be splitting hairs

System Operation—What it Does

The problem. The embassy cocktail party problem captures important features of the ocean surveillance mission. Suppose microphones are concealed for intercepting important conversations of foreign dignitaries at a party. Because bug emplacements are near the heavily trafficked bar and buffet tables, the basement sleuth who monitors the phones must contend with the babble of simultaneous speakers. The party-goers move into and out of range of the microphones in imperfectly predictable patterns. Their speaking patterns also change, as does the background clatter of plates, glasses, orchestra, and so on. Room echoes confound otherwise discernible conversations. Worst of all, the guests of greatest concern to the basement sleuth are prone themselves to covertness, using low voices and furtive movements in pursuit of their business.

HASP/SIAP sleuths in the deep ocean. Using data from concealed hydrophone arrays, it must detect, localize, and ascertain the type of each ocean vessel within range. The presence and movements of submarines are most important. Nevertheless, there are strategic and tactical motives for monitoring all vessel types.

Just as for the embassy sleuth, the program has to overcome the problems of non-cooperative subjects in a noisy, complex medium Ocean-going vessels typically move within fixed sea lanes, but storms and currents cause shifts in routes and operations. The background noise from distant ships is mixed with storm-induced and biological noises. Sound paths to the arrays vary with diurnal and seasonal cycles. Arrival of sound energy over several paths may suddenly shift to no arrivals at all, or arrivals only of portions of vessel radiation. Sound from one source can appear to arrive from many directions at once. Characteristics of the receivers can also cause sound from different bearings to mix, appearing to come from a single location. Finally, the submarine targets of most interest are very quiet and secretive.

What HASP/SIAP does to solve the problem. The program starts with digitized data from hydrophone arrays that monitor an ocean region from its periphery. The arrays have some directional resolution. Ideally each look direction produces a data channel with sound energy only from vessels near its axis, a spatial partition resembling spoke gaps on a bicycle wheel. In practice, radiation from a single vessel may spread across several gaps, and many vessels may be located in any one gap, or in adjacent gaps, a situation that can produce a kaleidoscope of sound.

Rotating shafts and propellers, and reciprocating machinery on board a ship are major sources of the intercepted radiation. The *signature*, or sound spectrum, of a ship under steady operation contains persistent fundamental narrowband frequencies and certain of their harmonics. Imagine the ship's propeller saying "ahhhhh" on its way across the ocean. On a speech analyst's sonogram, this sound would appear as a collection of dark vertical stripes against a fuzzy gray background.

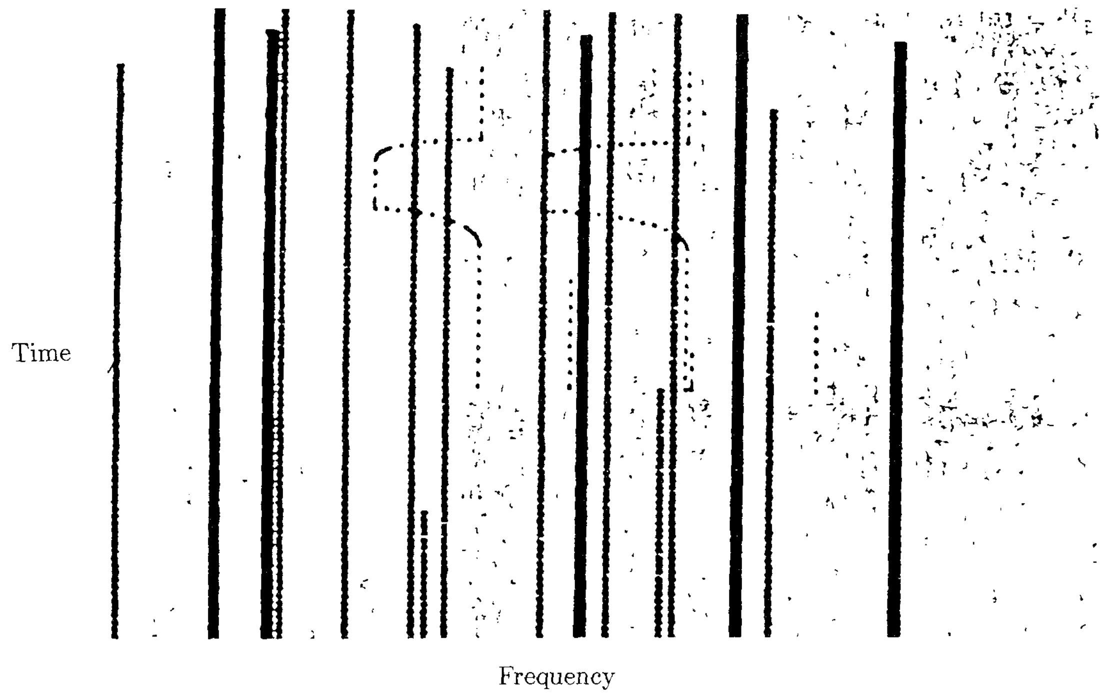


Figure 1. A simulated sonogram (with noise suppressed).

Sonar analysts have been trained to recognize the sound signature traits of ships on their sonogram displays, and to classify a signature into one of several important classes If only one ship is present on a channel, the problem is essentially to match the measured signature (it may be imperfectly measured or partially present) to a collection of stored references for the best fit. Most computer-aided ship classification schemes have achieved some measure of success at this level.

When several ships radiate into the same array channels, the problem becomes more demanding. Highly skilled analysts use a collection of tricks in sonogram interpretation to disentangle the signatures for classification Their procedures are not strictly separable for these tasks That is, they do not disentangle signatures first without regard to classification information.[3] HASP/SIAP is unique among current machine-aided classifiers in imitating this non-separable approach.

Sonogram displays used by a sonar analyst are analog histories of the spectrum of received sound energy. New data on a channel are portrayed by varying the intensity of pixels on the display. Greater concentrations of energy at a given frequency are translated into higher intensities at corresponding horizontal positions. Synchronous horizontal sweeps, therefore, leave vertical lines on a display where persistent frequencies are present. (See Fig 1 for a simulated sonogram.) Starting, stopping, frequency shifting, and even subtle traces, are discernible to a trained eye. Recognition of these analysis elements must also be carried out automatically if the program is to emulate human analysts' procedures in subsequent processing. The data streams from each hydrophone array channel are then converted into hypotheses on the class type of each detected vessel. The data, rules, and procedures associated with a particular hypothesis can be recalled, so supporting evidence for the program's conclusions are also available for operator scrutiny.

System Organization—How It Does It

Major terms and concepts. The *understanding* of sonograms often requires using information not present in the signals themselves. Major sources of information are reports from other arrays and intelligence reports. More general knowledge, like the characteristics of ships and common sea-lanes, also contributes significantly. Each such source of knowledge may at any time provide an inference which serves as a basis for another knowledge source to make yet another inference, and so on, until all relevant information has been used and appropriate inferences have been drawn.

Essential to the operation of the program is its *model* of the ocean scene. The model is a symbol-structure that is built and maintained by the program and contains what is known about the unfolding situation. The model thus provides a context for the ongoing analysis. More commonly known as the situation board to the analysts, the model is used as a reference for the interpretation of new information, assimilation of new events, and generation of expectations concerning future events. It is the program's *cognitive flywheel.*

[3] "Disentangle" means to separate correctly signature components of different vessels This process is aided by contextual information about the plausible behavior of the sources on ships

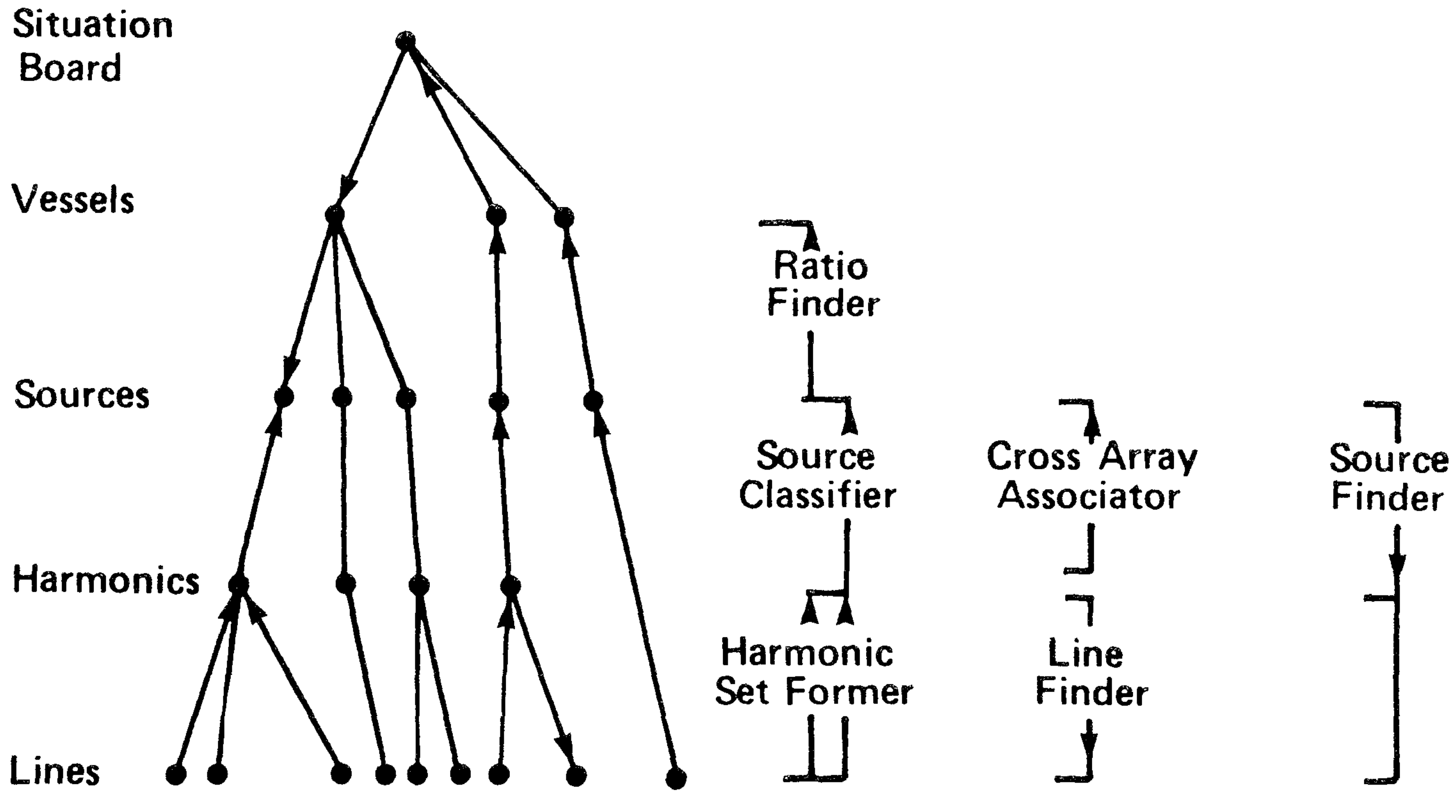


Figure 2. Levels of analysis and some of the knowledge sources

The task of understanding the situation from the sonogram and other data is accomplished at various *levels of analysis*. These levels are exhibited in Figure 2. The most integrated, or the highest, level represents the situation board describing all the ships hypothesized with some confidence. The lowest level, that is, the level closest to the data, consists of connected line segments containing features derived from the signal data. During the HASP design an assumption was made that a front-end program could be written to extract major signal features—intensity, stability, bandwidth, duration, etc. SIAP, in fact, integrates such a front-end signal processor into the system.

At each level, the *units of analysis* are the *hypothesis elements*. These are symbol-structures that describe what the available evidence indicates in terms that are meaningful at that particular level. Thus, on the Vessel level, in Figure 2, the descriptive properties that each Vessel element can have are Vessel Class, Location, Current speed, Course, and Destination. Each of the values of the properties has associated with it *weights*, an informal measure of confidence in the hypothesis. The example below shows a part of a hypothesis element on the Source level with different expressions of confidence.

```
SOURCE-1
    TYPE         (Engine .5) (Shaft 3) (Propeller – 3)
    LOCATION     ((Lat 34.2) (Long 126 5) (Error 9))
```

Links between the levels of analysis are built from *sources of knowledge*. A knowledge source (KS) is capable of putting forth the *inference* that some hypothesis elements present at its "input" level imply some particular hypothesis element(s) at its "output" level. A source of knowledge contains not only the knowledge necessary for making its own specialized inferences, but also the knowledge necessary for checking the inferences made by other sources of knowledge The inferences which draw together hypothesis elements at one level into a hypothesis element at a higher level (or which operate in the other direction) are represented symbolically as links between levels as shown in Figure 2. The resulting network, rooted in the input data and integrated at the highest level into a descriptive model of the situation is called the *current best hypothesis* (CBH), or the *hypothesis* for short.

Each source of knowledge holds a considerable body of specialized information that an analyst would generally consider "ordinary" Sometimes this is relatively "hard" knowledge, or "textbook" knowledge. Also represented are the *heuristics*, that is, "rules of good guessing" an analyst develops through long experience. These judgmental rules are generally accompanied by estimates from human experts concerning the *weight* that each rule should carry in the analysis.

Each KS is composed of "pieces" of knowledge. By a piece of knowledge we mean a *production rule*, that is, an IF–THEN type of implication formula. The "IF" side, also called the *situation* side, specifies a set of conditions or patterns for the rule's applicability The "THEN" side, also called the *action* side, symbolizes the implications to be drawn (or various processing events to be caused) if the "IF" conditions

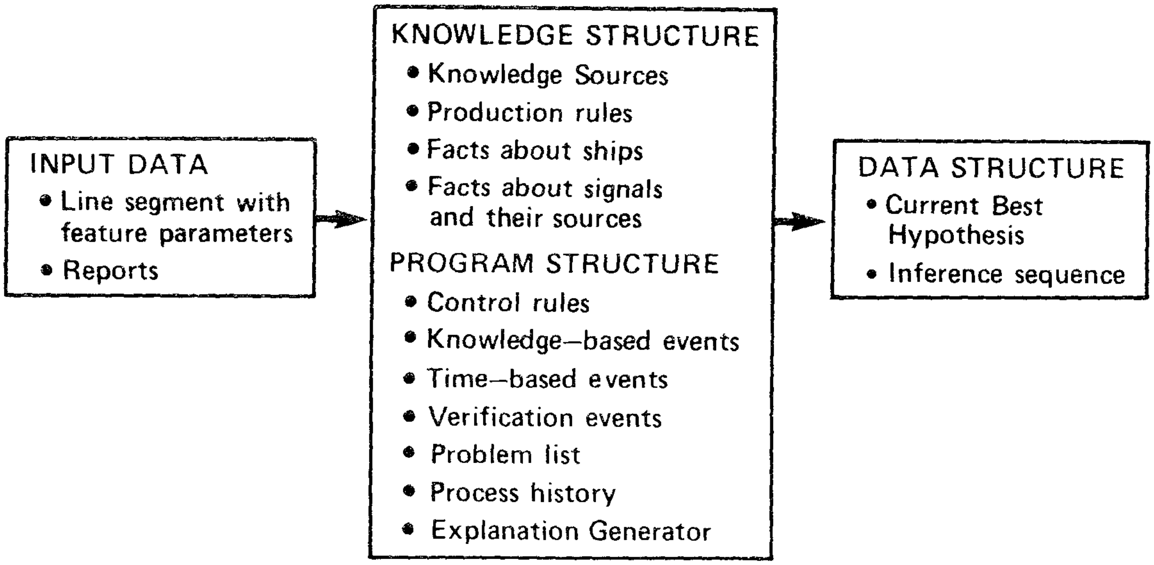


Figure 3 General program structure.

are met Following is a heuristic represented as a production rule:

IF Source was lost due to fade-out in the near-past, and
Similar source started up in another frequency, and
Locations of the two sources are relatively close,

THEN They are the same Source with confidence of 3

Source refers to some noise producing objects, such as propellers and shafts on ships

Hypothesis formation is an "opportunistic" process Both *data-driven* and *model-driven* hypothesis-formation techniques are used within the general *hypothesize-and-test* paradigm The *knowledge of how to perform*, that is, how to use the available knowledge, is another kind of knowledge that the analysts possess. This type of knowledge is represented in the form of *control rules* to promote flexibility in specifying and modifying analysis strategies. One of the tasks of the control knowledge source is to determine the appropriate techniques to use for different situations

The *unit of processing activity* is the *event*. Events symbolize such things as "what inference was made," "what symbol-structure was modified," "what event is expected in the future," and so on. The basic control loop for these *event-driven programs* is one in which lists of events and a set of control rules are periodically scanned to determine the *next thing to do*.

HASP/SIAP organization. Most signal processing programs are organized in a pipeline fashion starting with signal data, segmenting the signals, identifying the segments, and so on One way to view HASP/SIAP is as a signal processing paradigm with multiple feedbacks and many data input points. The primary input data is a sequence of *described line segments* (from each channel, for each array) present in the frequency vs time data. The secondary inputs are information available from other arrays and a variety of reports routinely available to the analysts. The output of the program is a data structure containing the program's best explanation of the current input data considered in conjunction with previous analyses of earlier-received data. This data structure is the machine equivalent of the analyst's situation board, except that it contains more information. In particular, it contains the basis for the explanation as recorded by the program during its analysis process. Figure 3 shows the general program structure of HASP/SIAP, and Figures 4 and 7 show some of the output.

Integration of many diverse data and knowledge sources is accomplished through a hierarchic data structure, as shown in Figure 2. The interpretation process is viewed as a problem of bidirectional, step-wise transformations between signals and the symbolic description of objects at various levels of abstraction, using as many intermediate steps as needed. Thus, for each level in the hierarchy, there must be at least one KS that can transform information on one level into information that is meaningful on some other level For example, the following rule transforms a line segment into a line by attaching it to a previously identified line:

IF Characteristics of a new segment "match" an earlier line, and
Source associated with the line is not currently heard, and
Source had disappeared less than 30 minutes ago,

THEN Source is being picked up again with confidence .5, and
Segment is assigned to the old line

The word "transformation" is used loosely to mean a shift from one representation of an object (e.g., signal segment) to another (e.g., propeller) using any formal or informal rules.

The current best hypothesis. As KSs are applied to a stream of input data, a solution hypothesis emerges The hypothesis structure represents the "best hypothesis" at any

```
The Current Best Hypothesis at time 20455

Vessel-1
Class        (OR (Cherry 8.4) (Iris 6.9) (Tulip 6.2) (Poppy 4.8) 20455 ...)
Location     ((Lat 37.3) (Long 123.1) (Error 37))
Speed        15.7
Course       135.9
Sources      (AND Source-1 Source-5)

  Source-1
  Type          (OR (Cherry Propeller 5.5) (Poppy Shaft 2.5)
                  (Poppy Propeller 2.0) (Cherry Shaft 2.5) 20455 ...)
  Dependency    Unknown
  Regain        (20230)
  Harmonics     (Harmonic-1)

    Harmonic-1
    Fundamental    (224 5 20520)
    Evolution      (fade-in 20230 fade-out 20210 ..)
    Lines          (AND Line-1 Line-2 Line-6 Line-12)

  Source-5
  Type          (OR (Cherry Shaft 6.0) (Poppy Shaft 4.0)
                  (Iris Propeller 5.0) (Tulip Propeller 2.0) 20455)
  Dependency    6
  Harmonics     (Harmonic-5)

    Harmonic-5
    Fundamental    (162.4 20455)
    Evolution      (fade-in 20455)
    Lines          (AND Line-25)

ASSIMILATION (RATIO Source-1 Source-5 .5) 20455)

Problem-list
(EXPECT Vessel-1 (SUPPORT Cherry) (Dependency Propeller 5))
(EXPECT Vessel-1 (PRED.LOC (Lat 37.2) (Long 123.) (Error 41.3))
(REPORT REPORT-GEN Rose (Signature (Engine 30 166.7) ..........))
```

The class of Vessel-1, located in the vicinity of Latitude 37.3 and Longitude 123.1 at time day 2, 4 hours, 55 minutes, can be either Cherry, Iris, Tulip, or Poppy class. Two distinct acoustic sources, supported by respective harmonic sets, have been identified for Vessel-1. Source-1 could be due to a shaft or propeller of vessel class Cherry or Poppy. Similar source possibilities exist for Source-5. These two sources were assimilated into Vessel-1 because of the possibility of a known mechanical ratio that exists between the two sources. If a dependency of the Cherry propeller for Source-1 can be determined to be 5, then the supporting evidence that Vessel-1 is a Cherry class can be increased. Additional information on the Problem-list suggests the expected position of Vessel-1 computed at the next time interval on the basis of its currently hypothesized location, course, and speed. In addition, there is a report that a Rose class vessel is expected in the area. (Also see Fig. 7 for a program-generated summary explanation.)

Figure 4. A part of a current best hypothesis.

given time for the data available up to that time. It is the most up-to-date situation board and contains all supporting evidence. The structure of the CBH is a linked network of nodes, where each node (hypothesis element) represents a meaningful aggregation of lower level hypothesis elements. A link between any two hypothesis elements represents a result of some action by a KS and indirectly points to the KS itself A link has associated with it directional properties. A direction indicates one of the following:

1 A link that goes from a more abstract to a less abstract level of the hypothesis is referred to as an *expectation-link*. The node at the end of an expectation-link is a model-based hypothesis element, and the link represents *support from above* (i e , the reason for proposing the hypothesis element is to be found at the higher level).

2. A link which goes in the opposite direction, from lower levels of abstraction to higher, is referred to as a *reduction-link* The node at the end of a reduction-link is a data-based hypothesis element, and the link represents *support from below* (i e , the reason for proposing the hypothesis element is to be found at a lower level). An example of hypothesis elements generated by the KSs are shown in Figure 4

Kinds of knowledge represented. There are several kinds of knowledge used in HASP/SIAP, each represented in a form that seems the most appropriate.

Knowledge about the environment: The program must know about common shipping lanes, location of arrays and their relation to map coordinates, and known maneuver areas. This knowledge is represented in procedures that compute the necessary information.

Knowledge about vessels: All the known characteristics about vessel types, component parts and their acoustic signatures, range of speed, home base, etc , are represented in frame-like structures. These constitute the static knowledge used by rules whenever a specific class of vessels is being analyzed. In addition, when some vessel class is inferred from a piece of data, detailed information is available to help make that hypothesis more credible. The use of this information by model-driven KSs reduces the amount of computation by directing other KSs to look for specific data

Interpretation knowledge: All heuristic knowledge about transforming information on one level of the CBH to another level is represented as sets of production rules The rules in a KS usually generate inferences between adjacent levels However, some of the most powerful KSs generate inferences spanning several levels For example, a line with particular characteristics may immediately suggest a vessel class. This type of knowledge is very situation-specific. It was elicited from human experts who know and use much of the specialized, detailed knowledge now in the program (It is said that chess masters can immediately recognize approximately fifty thousand board patterns.) There are more examples of rules in the next section.

Knowledge about how to use other knowledge: Since the strategy is opportunistic, the program must know when an opportunity for further interpretation has arisen and how best to capitalize on the situation. In HASP/SIAP this type of knowledge is made explicit as will be explained in the following section.

How the knowledge is organized and used. How well an Expert System performs depends both on the competence of the KSs and on the appropriate use of these KSs Thus, the primary considerations in the design of Expert Systems revolve around the availability and the quality of the KSs, and the optimal utilization of these KSs When and how a KS is used depends on its quality and its relevancy at any given time The relevance of a KS depends on the state of the CBH. The control mechanism for KS selection needs to be sensitive to, and be able to adjust to, the numerous possible solution states which arise during interpretation. Given this viewpoint, what is commonly called a "control strategy" can be viewed as another type of domain-dependent knowledge, albeit a high level one. Organizing the knowledge sources in a hierarchy is an attempt to unify the representation of diverse knowledge needed for the interpretation task

In a *hierarchically organized control structure*, problem solving activities decompose into a hierarchy of knowledge needed to solve problems. On the lowest level is a set of knowledge sources whose charter is to put inferences on the CBH We refer to KSs on this level as *Specialists*. At the next level there are *KS-activators* that know when to use the various Specialists. On the highest level a *Strategy-KS* analyzes the current solution state to determine what information to analyze next. The last activity is also known as *focussing-of-attention*

The execution cycle consists of

1 focussing attention on pending time-dependent activities, on verification of a hypothesis, or on one of the hypothesized elements;

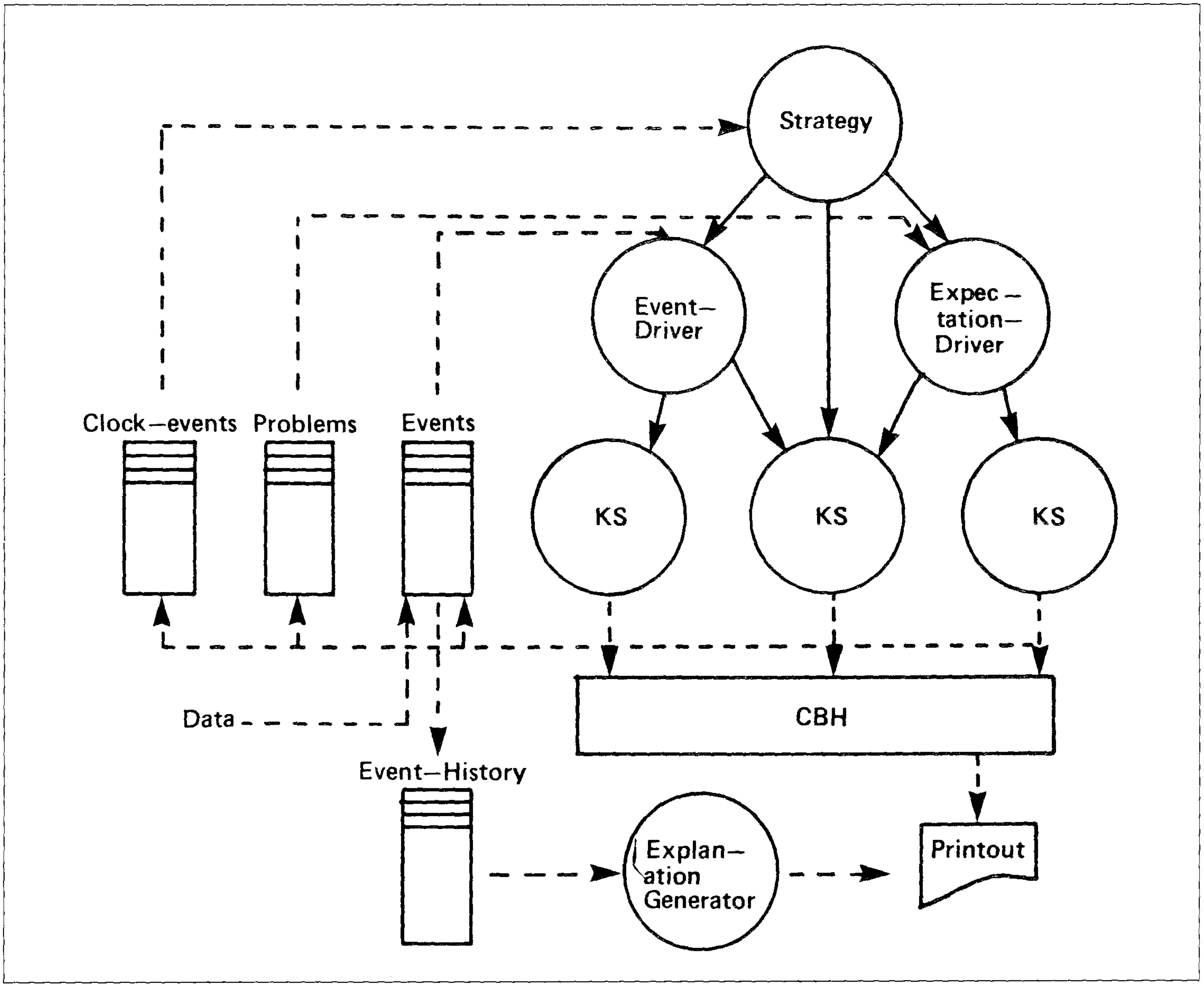
2 selecting the appropriate KSs for the attended event; and

3. executing the selected KSs

The KSs will generate a new CBH or expectations for some future events. The cycle is then repeated. Figure 5 shows the major flow of information and control. Since the program is event-driven, it is impossible to show the detailed flow of control in this type of a diagram.

The KS hierarchy should be clearly distinguished from the hierarchy of analysis levels The hypothesis hierarchy represents an *a priori* plan for the solution determined by a *natural* decomposition of the analysis problem. The KS hierarchy, on the other hand, represents a plan for organizing the problem-solving activities, or control, needed to form hypotheses. Figure 6 shows a general relationship between the organization of the hypothesis hierarchy and the KS hierarchy.

KSs on the Specialist level. Each Specialist has the task of creating or modifying hypothesis elements, evaluating inferences generated by other Specialists, and cataloging



Incoming data are treated as events and are put on the Event-list. At the begining of the processing cycle at time t, the items on the Clock-event list are scanned to see if it's time for any particular item to be processed If so, the appropriate KS is called. Next, the Expectation-driver is called to see if any pending problems can be resolved with the events on the Event-list. All resolvable problems are taken care of at this point. Next, the Event-driver is called to process the events. It first determines what changes of the CBH to focus on. (Note that each event represents either a new piece of data or a change made to the CBH during the last process cycle.) The Event-driver then calls the appropriate KSs to make further inferences on the basis of the focused event. The invoked KSs make changes to the CBH and add new events to the Event-list. A processing cycle terminates when the Event-list becomes empty.

Figure 5. Information and control flow.

missing evidences that are essential for further analysis. Its focus of attention is generally a hypothesis element containing the latest changes. Although a KS has access to the entire hypothesis, it normally "understands" only the descriptions contained in two levels, its input level and its output level. Some examples of different types of Specialists are listed below.

Inference-Generation:

IF Characteristics of a Harmonic set match another set on another channel,

THEN Both sets are coming from the same source with confidence of .6.

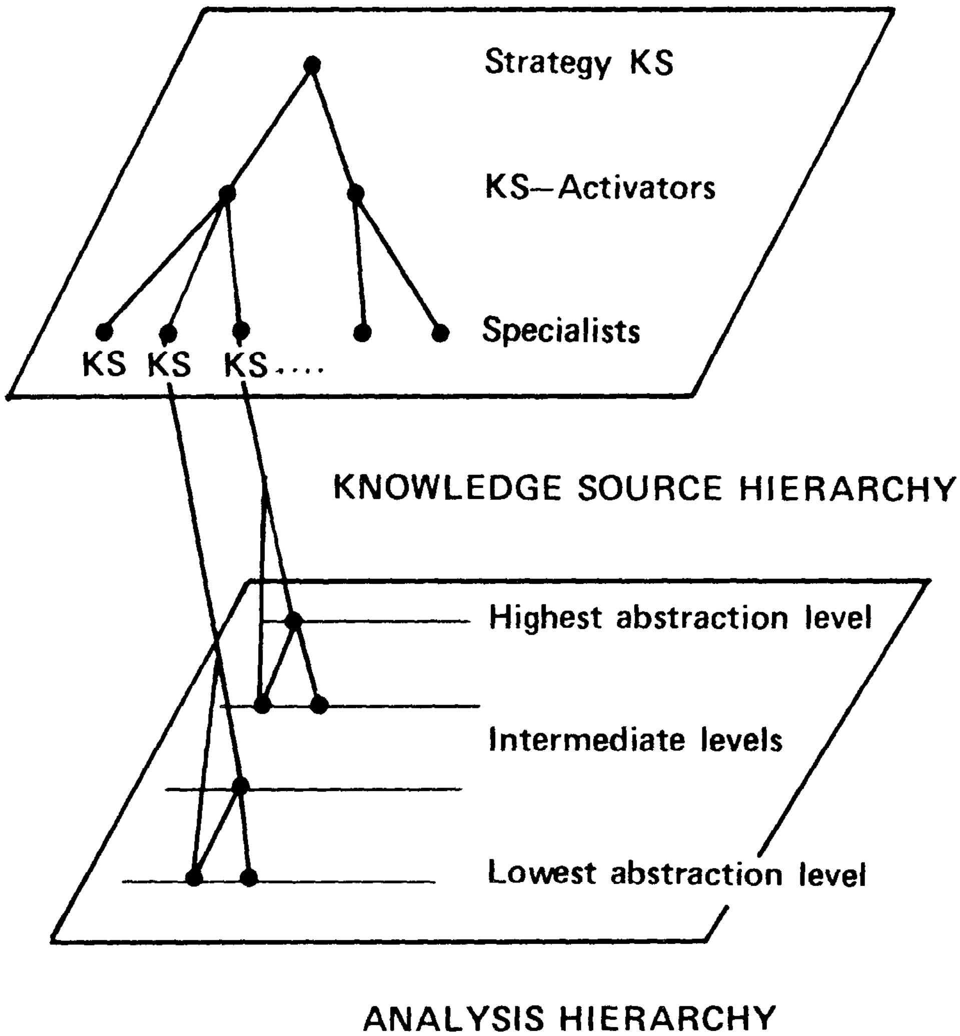


Figure 6. Relationship between the analysis levels and the KS levels.

Inference-Evaluation:

IF Source belongs to Vessels of class Cherry or Iris, and Harmonics associated with Source have been stable for a while,
THEN Increase the confidence of Cherry and Iris by .3

Problem-Cataloging:

IF Report exists for a vessel class Rose to be in the vicinity, and
Source likely to be associated with Rose has been detected,
THEN Expect to find other Source types associated with Rose class.

KSs at the KS-activation level. A KS on this level has the task of invoking the appropriate Specialists given the kind of strategy being employed. For example, a KS charged with calling the appropriate KSs within a model-driven strategy has a different goal than one charged with a data-driven strategy. Different KS-activators can be made to reflect different policies, ranging from using the fastest-first to the most-accurate-first. HASP/SIAP has two KS-activators, the Event-driver and the Expectation-driver. If there is more than one Specialist available to process an event, some policy is needed to guide the order in which these KSs are to be used. The Event-driver chooses items on the Event-list and activates Specialist-KSs based on the degree of specialization (and assumed accuracy) of the KSs. The Expectation-driver processes items on the Problem-list on the basis of how critical the needed evidence is to the emerging hypothesis.

Event-driver: An event type represents an *a priori* grouping of similar changes to the hypothesis (i.e. it represents the abstractions of possible changes to the hypothesis). An example event is "source-type-identified." The changes, together with the identity of the rules that produced the changes, are put on a globally accessible list called the *Event-list.* The Event-driver invokes the appropriate Specialist-KSs based on the focused event or group of events.

Expectation-driver: The Expectation-driver monitors the items on the Problem-list to see if any event that might satisfy an expectation on the Problem-list has occurred. If an expected event does occur, the Expectation-driver will call the Specialist-KS which put the expectation on the Problem-list. For example, in Figure 4, if a source belonging to a reported Rose class is detected, then REPORT-GEN (the KS that put the item on the Problem-list) will be called.

KS on the Strategy level. The Strategy-KS reflects a human analyst's problem-solving strategy. Its expertise consists of determining how accurate the current hypothesis is and in deciding what task will have the highest impact on the CBH. It has a variety of possible activities to choose from:

"Is it time to check for specific data?"

"Has anything happened since the last processing cycle that might invalidate the hypothesis?"

"Has an expected event occurred?"

"What is the best region in the CBH to work on next (i.e., what is the region of minimal ambiguity)?"

In HASP/SIAP there are no formal mechanisms to measure the differences between the current best hypothesis and the "right answer." The program detects when the solution hypothesis is "on the right track" by a periodic use of heuristic criteria. A consistent inability to verify expectation-based hypothesis elements may signal an error in the hypothesis. A more general indication of ineffective hypothesis formation appears as a consistent generation of conjectures whose confidence values are below a threshold value; and which therefore indicates that the analysis is "bogged down."

Dealing with time-dependent analysis. The task of HASP/SIAP is to interpret continuous streams of data and to maintain a current situation model. The primary input data currently consists of 5-minute segments describing, in effect, a summary of observed signals at various frequencies. These segments must be integrated into the existing Line-level elements. In addition, lines must be integrated into harmonic sets; sources must be attributed to vessels. The CBH serves as the historical context by which the integration can occur. Through the CBH one can incorporate appearances and disappearances of signals over time, using only common-sense reasoning, such as, "ships can't just disappear, or appear, from nowhere" The CBH must keep track of the times when specific events occurred, as well as maintain a network of relationships between Lines, Sources, and Vessels. Time markers are placed with hypothesized values; new time markers are added only when the values change. In the example below, there were no changes to the Source type for two hours even though weights may have been changed during that time. (Also see Fig. 4.)

```
SOURCE-1
    TYPE [(OR (Cherry Propeller 7 5)) 10650
          (OR (Cherry Propeller 6 0)
          (Poppy Propeller 2.5)) 10450]
```

HASP/SIAP must also analyze time-oriented signatures and movements of vessels over time. The signature analysis is similar to trend analysis. The signature analysis depends on the type of vessel—some vessel signatures may last 15-minutes, others may last for hours or days. This type of time-dependent analysis is accomplished through the use of a mechanism called *Clock events.* When a Specialist-KS needs to look at or verify signatures, it puts on the Clock-event list a request to be recalled at a specific time. The request contains information that needs to be reviewed at that time. The Strategy-KS also generates Clock events in order to periodically review certain hypothesis elements at specified time intervals. In the example below a Specialist-KS called RATIO-FINDER will be called at time 22055 to check for a ratio between the Shaft and the Propeller for the Cherry class hypothesis.

Clock Event
(22055 RATIO-FINDER Source-1 Source-5
(Cherry Shaft Propeller))

Explanation and summary of the CBH. Every so often an explanation of the current best hypothesis is printed, together with unresolved issues. There were difficulties in generating an explanation that the analysts could understand. Although the basic processing strategy of the program had to be breadth-first, the analysts wanted to know all the supporting rationale for the situation one vessel at a time, that is, depth-first. Furthermore, analysts were not interested in every inference step, only the "important" ones. These requirements eliminated the use of a straightforward back-trace explanation currently in use in most Expert Systems. Another knowledge source had to be brought to bear on the History-list to identify "important" events and to present these events in a sequence acceptable to the analysts. Figure 7 contains a small sample of a HASP/SIAP summary explanation. The first statement requires a search through the History-list to collect several events that support the Vessel-1 hypothesis. These events are widely separated events in the processing cycle.

AI and Signal Processing

Decomposition of the HASP/SIAP program offers an interesting avenue for highlighting the value of combining AI with signal processing. The two segments of importance are the following:

1 Translation of the data into sonogram lines and measurement of line parameters; and

2 The disentanglement of intermingled target signatures and their classification.

The HASP system used skilled human perception to perform the first step in the signal-to-symbol transformation—the encoding of the signal data into segments and features For a largely automated classification program this is a task

Summary of Events at this Time Slot

- Newly reported line segment which does not seem to be a part of any previously observed lines is assigned to a new line: Line-25. This line is assigned to a new lineset Harmonic-5, from which in turn a new source Source-5 is created.
- Sources Source-1 and Source-5 of vessel Vessel-1 have a known ratio; this provides further evidence for the association of these two sources with this vessel.
- Source type of Source-5 is updated due to an observed ratio: (Cherry Shaft 6.0) (Poppy Shaft 4.0) (Iris Propeller 5.0) (Tulip Propeller 2.0).
- Source type of Source-1 is updated due to an observed ratio: (Cherry Propeller 4.0).
- Based on modified Source types (Source-1), the class type of Vessel-1 are weighted as follows: (Cherry 7.2).
- Based on modified Source types (Source-5), the class types of Vessel-1 are weighted as follows: (Cherry 8 4) (Iris 6.9) (Tulip 6 2) (Poppy 4.8).

Pending Problem

There is evidence that Vessel-1 is a Cherry class, because there is a relation between Shaft and Propeller.

If it is a Cherry class, there should be a dependency of 5 in the propeller and dependency of 6 in the shaft. We have not yet observed the dependency 5.

Problem list = (EXPECT Vessel-1 (SUPPORT Cherry) (Dependency Propeller 5))

A summary explanation is printed with the current best hypothesis, shown in Figure 4. The CBH describes the current situation; the summary explanation provides the rationale for the changes in the situation since the last processing cycle.

Figure 7 Example of a summary explanation.

that requires substantial signal processing. Complex algorithms have now been developed for this front-end application, incorporating many of the skills of the analyst.

However, in the SIAP project the sponsor required that a particular front-end be used. This program unfortunately smoothed over many of the key clues that permitted the HASP program to perform so well. To compensate for this setback, some AI had to be moved into the signal processing front-end and the processing adjusted in the second segment. For example, the automated front-end had a tendency to break certain wideband lines into several adjacent narrowband lines. An uninformed grouping of these lines can be incorrect, since in some cases they could be neighboring narrowband lines. The grouping needs to be accomplished within the context of what is already known about the relationship among the lines, and between the lines, the sources, and the vessel classes. A new knowledge source was added that used the context to process the algorithmic outputs.

Two other major issues involving AI and signal processing surfaced in the course of building HASP/SIAP They are:

1. the feedback to the signal processing front-end; and
2. the allocation of signal processing resources

There are two kinds of feedback that the situation model can provide to signal processors. First, special purpose detection and parameter measurement algorithms depend on higher level information—the CBH has the necessary information. Second, threshhold values for the front-end need to be adjusted according to the current needs and expectations. In both cases, the processing of information over a period of time leads to modifications of parameters in the front-end for more accurate subsequent processing.

In the undersea surveillance problem, a variety of signal processing techniques can be brought to bear in the front-end. Some of these fall in the category of "standard" algorithms that are more or less always used. But others are specialized algorithms that cannot be used for all channels at all times because of their cost. Because these algorithms can provide important processing clues when used, the allocation of these scarce resources is an important issue. The appropriate use of the resource is especially important because the use of these special algorithms can significantly reduce the time required to produce an accurate situation model

The resource allocation problem is knowing *when* to invoke the special signal processors. The approach to its solution lies in rules that can recognize when the context will permit the special processor to resolve important ambiguities. In HASP/SIAP only a rudimentary capability for this process was used, but its value was conclusively demonstrated.

The resource allocation problem pointed out another "signal processing" capability that can be utilized on a demand basis to resolve ambiguities. This resource is the human operator, who generally has great expertise in the entire process, including the processing of the sonogram information. This interaction was performed off-line in HASP/SIAP, but on-line capabilities are being incorporated in the HASP/SIAP derivatives. This approach is more in line with using the Expert System as an analyst aid, rather than a stand-alone system, but the combination of man and machine will be far more powerful than either alone.

The HASP/SIAP experience indicates that some good AI can cover a multitude of signal processing inadequacies and can direct the employment of signal processing algorithms. The intelligent combination of AI and signal processing views the signal processing component as another knowledge source, with rules on how best to employ algorithms and how to interpret their output.

Evaluation

MITRE Corporation undertook a detailed analysis of the performance of SIAP in a series of experiments conducted between December 1976 and March 1978 at DARPA's Acoustic Research Center. The experiments compared HASP/SIAP performance with those of two expert sonar analysts in three task categories.

The first experiment, designed to test the performance in detection and classification of vessels using data derived from actual ocean events, led to the general conclusion that "HASP/SIAP has been shown to perform well on ocean derived data... For this restricted ocean scene, the program is not confused by extraneous data and gives results comparable to an expert analyst."

The second experiment, designed to test the information integration capability, using derived data for multi-arrays, led to the conclusion that "HASP/SIAP understood the ocean scene more thoroughly than the second analyst and as well as the first analyst... The program can work effectively with more than one acoustic array. SIAP classified an ocean scene over a three hour time period indicating the plausibility of SIAP efficacy in an evolving ocean situation."

The third and final experiment, documented by MITRE, was designed to test the automatic parameter extraction capability added during the SIAP phase of HASP/SIAP development. It led to the conclusion that "with the exception that the SIAP program obtained significantly more contacts than the human analysts, the descriptions of the ocean scene are very similar." Moreover, "SIAP can perform vessel classification in increasingly difficult ocean scenes without large increases in the use of computer resources." Hence, continued better-than-real-time performance could be expected if the system were deployed

In a later experiment, it was shown that the additional contacts seen by SIAP in the third experiment were due to the front-end processor provided by the sponsor—namely, taking relatively wideband lines and decomposing them into erratic collections of narrowband lines. These problems were essentially eliminated by additional heuristics in the line-formation knowledge source.

Conclusions

In signal processing applications, involving large amounts of data with poor signal-to-noise ratio, it is possible to reduce computation costs by several orders-of-magnitude by the use of knowledge-based reasoning rather than brute-force statistical methods We estimate that HASP/SIAP can reduce computation costs by 2 to 3 orders-of-maginitude over conventional methods. *It makes little sense to use enormous amounts of expensive computation to tease a little signal out of much noise, when most of the understanding can be readily inferred from the symbolic knowledge surrounding the situation.*

There is an additional cost saving possible. Sensor bandwidth and sensitivity is expensive. From a symbolic model it is possible to generate a set of signal expectations whose emergence in the data would make a difference to the verification of the ongoing model. Sensor parameters can then be "tuned" to the expected signals and signal directions; not every signal in every direction needs to be searched for.

Suitable application areas. Building a signal interpretation system within the program organization described above can best be described as *opportunistic* analysis. Bits and pieces of information must be used as opportunity arises to build slowly a coherent picture of the world—much like putting a jigsaw puzzle together. Some thoughts on the characteristics of problems suited to this approach are listed below.

1. *Large amounts of signal data need to be analyzed.* Examples include the interpretation of speech and other acoustic signals, x-ray and other spectral data, radar signals, photographic data, etc A variation involves understanding a large volume of symbolic data; for example, the maintenance of a global plotboard of air traffic based on messages from various air traffic control centers.

2 *Formal or informal interpretive theories exist.* By informal interpretive theory we mean *lore* or heuristics which human experts bring to bear in order to *understand* the data These inexact and informal rules are incorporated as KSs in conjunction with more formal knowledge about the domain.

3 *Task domain can be decomposed hierarchically in a natural way* In many cases the domain can be decomposed into a series of data reduction levels, where various interpretive theories (in the sense described above) exist for transforming data from one level to another

4 *"Opportunistic" strategies must be used.* That is, there is no computationally feasible *legal move generator* that defines the space of solutions in which pruning and steering take place Rather, by reasoning about bits and pieces of available evidence, one can incrementally generate partial hypotheses that will eventualy lead to a more global solution hypothesis.

Data-driven vs. model-driven hypothesis-formation methods. Data- and model-driven methods of hypothesis formation were combined in the design of HASP/SIAP. By *data-driven* we mean "inferred from the input data." By *model-driven* we mean "based on expectation" where the expectation is inferred from knowledge about the domain. For example, a hypothesis generated by a KS which infers a source type from a harmonic set is a data-driven hypothesis. On the other hand, a hypothesis about the possible existence of a harmonic set based on the knowledge about a ship is a model-based hypothesis. In the former case, the data are used as the basis for signal analysis; in the latter case, the primary data are used solely to verify expectations.

There are no hard-and-fast criteria for determining which of the two hypothesis formation methods is more appropriate for a particular signal-processing task. The choice depends, to a large extent, on the nature of the KSs that are available and on the power of the analysis model available. Our experience points strongly toward the use of a combination of these techniques; some KS's are strongly data dependent while others are strongly model dependent. In HASP/SIAP the majority of the inferences are data-driven, with occasional model-driven inferences. The following are guidelines we have used in the past to determine which of the two methods is more appropriate:

1. *Signal-to-noise ratio* Problems which have inherently low S/N ratios are better suited to solutions by model-driven programs; the converse is true for problems with high S/N ratios. However, designers should beware that the model-driven approach is prone to "finding what is being looked for " Model-driven approaches should be supplemented with strong verification heuristics.

2. *Availability of a model* A model, sometimes referred to as *the semantics of the task domain,* can be used in various ways: (a) as input at some level of the hypothesis structure, (b) to make inferences based on general knowledge about the task domain, or (c) to make inferences based on specific knowledge about the particular task In HASP/SIAP, the model is drawn from general knowledge about the signal sources and from external reports that serve to define the context. If a reliable model is available, the data-interpretation KSs can be used as *verifiers* rather than *generators* of inferences; this reduces the computational burden on the signal-processing programs at the "front end."

The general methodology used in HASP/SIAP have been applied in many other problem areas. A small sampling includes the HEARSAY-II speech understanding program, the original Blackboard program (Erman, Hayes-Roth, Lesser, and Reddy, 1980; Lesser and Erman, 1977); CRYSALIS, a program that interprets protein x-ray crystallographic data (Engelmore and Terry, 1979); a program that generates plans (Hayes-Roth and Hayes-Roth, 1979); a program that interprets aerial photographs (Nagao, Matsuyama, and Mori, 1979). In addition, program packages that help users implement a variety of Blackboard programs have been under development for the past few years (Erman, London, and Fickas, 1981; Nii and Aiello, 1979).

References

Engelmore, R. S , & Terry, A (1979) Structure and function of the CRYSALIS system. *IJCAI 6,* 251–256.

Erman, L. D., Hayes-Roth, F., Lesser, V. D , & Reddy, R D (1980) The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys* 12:213–253

Erman, L. D , London, P. E , & Fickas, S F. (1981) The design and an example use of HEARSAY-III. *IJCAI 7,* 409–415

Feigenbaum, E. A. (1977) The art of artificial intelligence: I Themes and case studies of knowledge engineering *IJCAI 5,* 1014–1029.

Feigenbaum, E. A (1980) Knowledge engineering: The applied side of artificial intelligence. Tech Rep. STAN-CS-80-812, Computer Science Dept., Stanford University

Hayes-Roth, B., & Hayes-Roth, F (1979) Modeling planning as an incremental, opportunistic process. *IJCAI 6,* 375–383

Lesser, V. F , & Erman, L D. (1977) The retrospective view of the HEARSAY-II architecture. *IJCAI 5,* 790–800.

Lindsay, R , Buchanan, B. G , Feigenbaum, E A , & Lederberg, J (1980) *Applications of artificial intelligence for organic chemistry: The DENDRAL project.* New York: McGraw-Hill

Nagao, M., Matsuyama, T , & Mori, H. (1979) Structured analysis of complex photographs. *IJCAI 6,* 610–616.

Newell, A., Barnett, J., Green, C., Klatt, D , Licklider, J C R , Munson, J., Reddy, R., & Woods, W (1973) *Speech understanding systems: Final report of a study group* Amsterdam: North-Holland

Newell, A , & Simon, H. A (1976) Computer science as empirical inquiry: Symbols and search *Communications of the ACM* 19:113–126.

Nii, H. P., & Aiello, N. (1979) AGE: A knowledge-based program for building knowledge-based programs. *IJCAI 6,* 645–655.

Nii, H P., & Feigenbaum, E. A (1978) Rule-based understanding of signals. In D A. Waterman & F. Hayes-Roth (Eds), *Pattern-directed inference systems.* New York: Academic Press