Distributed Interpretation: A Model and Experiment

VICTOR R. LESSER AND LEE D. ERMAN

Abstract—The range of application areas to which distributed processing has been applied effectively is limited. In order to extend this range, new models for organizing distributed systems must be developed.

We present a new model in which the distributed system is able to function effectively even though processing nodes have inconsistent and incomplete views of the databases necessary for their computations. This model differs from conventional approaches in its emphasis on dealing with distribution-caused uncertainty and errors in control, data, and algorithm as an integral part of the network problem-solving process.

We will show how this new model can be applied to the problem of distributed interpretation. Experimental results with an actual interpretation system support these ideas.

Index Terms—Cooperative problem solving, distributed artificial intelligence, distributed interpretation, distributed processing, knowledge-based interpretation system.

Manuscript received January 10, 1980; revised July 9, 1980. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract F44620-73-C-0074 to Carnegie-Mellon University, Pittsburgh, PA, the National Science Foundation under Grant MCS78-04212 to the University of Massachusetts, Amherst, and DARPA under Contract DAHC 1572-C-0308 to the University of Southern California, Marina del Rey. The views and conclusions contained in this document are the authors' and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any other person or agency connected with them. An expanded version of this paper appeared in *Proc. 1st Int. Conf. on Distributed Comput. Syst.*, Huntsville, AL, October 1979, under the title "An Experiment in Distributed Interpretation."

V. R. Lesser is with the Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

L. D. Erman is with the Information Sciences Institute, University of Southern California, Marina del Rey, CA 90291.

I. Introduction

An interpretation system accepts a set of signals from some environment and produces higher level descriptions of objects and events in the environment. Speech and image understanding, medical diagnosis, determination of molecular structure, and geological surveying are problems that have been pursued with interpretation systems. A *distributed* interpretation system may be needed for applications in which sensors for collecting the environmental data are widely distributed, interpretation requires data from at least several of the sensors, and communication of all sensory data to a cen-

0018-9340/80/1200-1144$00.75

tralized site is undesirable. Sensor networks (composed of low-power radar, acoustic, or optical detectors, seismometers, hydrophones, etc.), network (automotive) traffic control, inventory control (e.g., car rentals), power network grids, and tasks using mobile robots are examples of potential applications for distributed interpretation. In these applications, an architecture that locates processing capability at the sensor sites and requires only limited communication among the processors is especially advantageous and is, perhaps, the only way to meet the demands of real-time response, limited communication bandwidth, and reliability.

Two major questions arise in the distributed interpretation task: how to interpret the signal data and how to decompose a given interpretation technique for distribution. Some interpretation algorithms and control structures cannot be replicated or partitioned on the basis of the distribution of the sensory data without requiring unacceptably large amounts of interprocessor communication to maintain completeness and consistency among the local databases. In such a case, it is necessary to modify the algorithm and control structure to operate on local databases that are incomplete and possibly inconsistent. For some interpretation techniques, such modifications might be difficult or impossible.

Knowledge-based artificial intelligence (AI) interpretation systems developed recently for speech, image, and signal interpretation applications have structures that seem to make them suitable for decomposition in distributed environments where local databases are incomplete and possibly inconsistent. Examples of these systems include Hearsay-II [6], HARPY [18], MSYS [1], SIAP [3], CRYSALIS [4], and VISIONS [10]. These interpretation techniques use the problem-solving paradigm of searching for an overall solution by the *incremental aggregation of partial solutions*. In this paradigm, errors and uncertainty from input data and incomplete or incorrect knowledge are handled as an *integral* part of the interpretation process. This is in contrast to more conventional problem-solving techniques, in which errors are fatal or are handled as exceptional conditions, requiring additional processing outside the normal problem-solving strategy.

We hypothesize that these knowledge-based AI systems can handle the additional uncertainty introduced by a distributed decomposition without extensive modification.[1] Preliminary work in testing this hypothesis with respect to synchronization has been encouraging. Experiments with a multiprocessor implementation of the Hearsay-II speech-understanding system have shown that eliminating explicit synchronization results in increased parallelism without a decrease in problem-solving accuracy [7]. Similarly, a class of iterative refinement methods (although not knowledge-based) for solving partial differential equations has been decomposed for multiprocessor implementation so as to avoid most explicit synchronization, thus allowing for increased speed-up due to parallel processing [2]. This decomposition is accomplished by allowing each point in the differential grid to be calculated from values of its neighboring points that are not necessarily the most up-to-date.

While such AI systems provide a promising basis for distributed problem solving, none has yet been built for a fully distributed environment; centralized global knowledge or global control has been used in existing interpretation systems to coordinate various system modules. In this report, we describe an experiment in the complete decomposition of an existing knowledge-based interpretation model—Hearsay-II [5], [15]. Although Hearsay-II was developed in the context of speech understanding [14], [6], its basic structure has been applied to a range of interpretation tasks, including multisensor signal interpretation [19], protein-crystallographic analysis [4], and image understanding [10].

This report concentrates on applying the Hearsay-II architecture to the distributed interpretation problem, where each processor can be mobile, has a set of (possibly nonuniform) sensing devices, and interacts with nearby processors through a packet-radio communication network [13]. Processors communicate among themselves to generate a consistent interpretation of "what is happening" in the environment being sensed.

Section II presents a brief overview of the Hearsay-II model of knowledge-based AI interpretation, followed by a description of the Hearsay-II architecture. This section presents mechanisms for handling uncertainty as an integral part of the problem-solving process. Section III outlines several possible directions for designing a distributed Hearsay-II architecture, with Section IV presenting the particular organization we feel most appropriate.

Section V describes the details of a distributed Hearsay-II speech-understanding system based on this organization. Each node is a functionally complete Hearsay-II system with access to one segment of the speech input data of the utterance. The nodes cooperatively generate an interpretation of the entire utterance by communicating partial, tentative interpretations based on their local views. Section VI presents experimental performance of this distributed speech-understanding system and compares it to that of the centralized system. This includes comparisons of several internode communication strategies, as well as the effects of communication errors. We discuss here and in Section VII how the Hearsay-II mechanisms are able to resolve successfully with low-internode communication the uncertainty introduced by the distribution of the system.

Our goal is not to prove that one *should* design a distributed speech-understanding system, but rather to point out some of the issues involved in designing a distributed interpretation system dealing with incomplete and inconsistent local data as an integral part of its processing. We are using the Hearsay-II speech-understanding system because it has a structure that we feel is appropriate and because it is a large, knowledge-based interpretation system to which we have access. There are serious problems with using this system for experimentation.

1) Because of several considerations discussed in Sections V-B and VI-A, networks are limited to about three nodes.

2) Because of the costs of the network simulation, only a

[1] A more detailed discussion of these points and the appropriateness of knowledge-based AI as the basis for distributed problem-solving systems is contained in [17].

limited number of experimental runs could be done and with relatively simple test data and communication policies.

3) There is probably no practical need for distributing a single-speaker speech-understanding system.

We feel that these limitations are sufficiently outweighed by the advantages of experimentation with a *real* system to make the effort worthwhile and the results, while not conclusive, indicative.

II. Overview of Hearsay-II: A System That Handles Uncertainty

A. The Model

We will take, as the competence goal of an interpretation system, the construction of the most credible complete interpretation of the input data.[2] In Hearsay-II, an interpretation is constructed by combining partial interpretations derived from diverse knowledge. Each area of knowledge is represented by an independent module called a "knowledge source" (*KS*). In the application of Hearsay-II to speech understanding, for example, these *KS*'s cover such knowledge areas as acoustics, phonetics, syntax, and semantics. The Hearsay-II architecture is designed to permit cooperative and competitive problem solving among the *KS*'s in order to resolve the uncertainty caused by noise and incompleteness in the input data and inaccurate processing by the *KS*'s.

The interaction of *KS*'s is based on an iterative data-directed form of the hypothesize-and-test paradigm. In this paradigm, an iteration involves the creation of an hypothesis, one possible interpretation of some part of the solution, followed by test(s) of its plausibility. When performing these actions, *KS*'s use *a priori* knowledge about the problem, as well as previously generated hypotheses that form a context for applying the knowledge. When a *KS* creates an hypothesis from previously created hypotheses, the *KS* extends the existing (partial) interpretation with more information, thereby reducing the uncertainty of the interpretation. The processing is terminated when a consistent hypothesis is generated that satisfies the requirements of a complete solution.

A *KS* often generates incorrect hypotheses because its knowledge or its input data, including previously generated hypotheses, contains errors or is incomplete. Thus, if *KS*'s were to generate only a single hypothesis for each specific part of the problem, the problem-solving process would often terminate with an inaccurate interpretation or with a partial interpretation that could not be extended because of its inconsistency. In order to avoid this problem, *KS*'s, in general, create several *alternative* hypotheses for each part of the problem. The *KS* associates with each hypothesis a *credibility* rating, which is its estimate of the likelihood that the hypothesis is correct. The lower the credibility of the alternatives, the greater the number that must be generated to produce the same likelihood that a correct one is included.

[2] In general, some applications might not contain a notion of a complete or spanning interpretation, but rather are interested in successive partial interpretations. Nothing in the discussion that follows is actually specific to complete interpretations, but we adopt that notion because of our involvement with the speech-understanding task and the interpretation of individual single-sentence utterances.

The set of all possible partial interpretations defines the problem-solving search space. The more alternative hypotheses generated, the larger the fraction of the space actually searched. Since each partial interpretation can give rise to multiple extensions, the possibility of a combinatoric explosion exists. At each step in the search, a subset of the existing partial interpretations is selected for extension; the resulting extended partial interpretations then compete for selection with those previously generated. The selection of the subset of hypotheses to extend is called the *focus-of-control* (or focus-of-attention) problem. An integral part of effective focus-of-control is the problem-solving system's ability to focus quickly on information that constrains the search, in order to contain combinatoric explosions. This is called an *opportunistic* and *asynchronous* style of problem solving. It can be implemented through the Hearsay-II formulation of the hypothesize-and-test paradigm, in which promising tentative decisions are made (despite incomplete information or knowledge), then reevaluated later in the light of new information. Focus-of-control is discussed further below; it is also discussed more extensively in [11].

Three requirements must be met for the effective operation of this general approach to problem solving.

1) *Sufficiency of Knowledge:* The knowledge can generate some sequence of partial interpretations that culminates in a correct complete interpretation.

2) *Sufficiency of Credibility Evaluation:* The credibility function rates the correct complete interpretation higher than any incorrect complete intepretation generated.

3) *Sufficiency of Control Strategy:* The focus-of-control strategy can find a correct complete interpretation within the bounds of computing resources allocated to the task.

Increasing the constraint of knowledge, the discrimination power of the credibility evaluation or the selectivity of the control strategy beyond that which is minimally sufficient to meet these criteria will, in general, decrease the amount of computing resources needed for the interpretation. Also, these three aspects of the problem solving are not independent; within limits, the same performance can be achieved by trading off the uncertainty resolving power of one aspect for that of another.

B. The Architecture

Fig. 1 shows a simplified schematic of the centralized Hearsay-II architecture. The major data structures are the shared global database (called the *blackboard*), focus-of-control database, and scheduling queues.

The blackboard is partitioned into distinct information levels, each used to hold a different kind of representation of the problem space. The major units on the blackboard are the hypotheses. Relationships among hypotheses at different levels are represented by a graph structure. The sequence of levels on the blackboard forms a loose hierarchical structure in which the elements at each level can be described approximately as abstractions of elements at the next lower level. For example, in speech understanding an utterance can be represented as a signal or as sequences of phones, syllables, words, phrases, or concepts; in image understanding, typical levels might in-

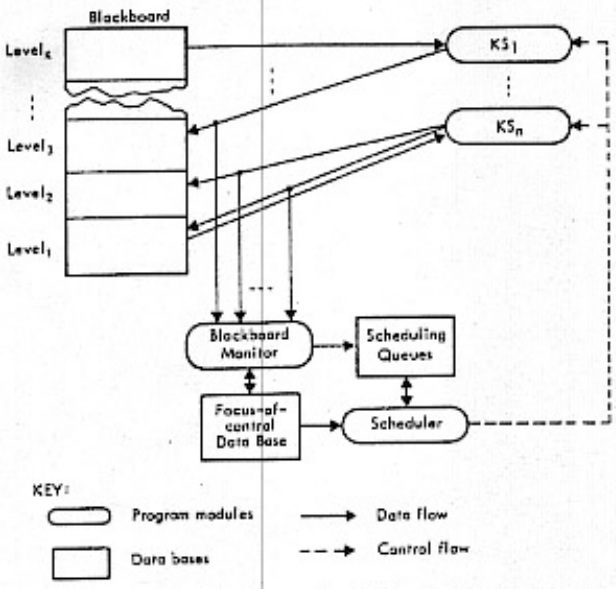

Fig. 1. Schematic of the (centralized) Hearsay-II architecture.

clude picture points, line segments, areas, surfaces, and objects; levels in an aircraft-tracking radar system might include signals, signal groups, vehicles, area maps, and overall area maps (see [21]). The set of possible hypotheses at a level forms a problem space for *KS*'s operating at that level. A partial interpretation (i.e., a group of hypotheses) at one level can be used within the opportunistic strategy to constrain the search at another level. For example, a *KS* can create a phrase hypothesis as an abstraction of a sequence of word hypotheses. Similarly, another *KS* can use the phrase hypothesis to predict (i.e., constrain) the set of possible word hypotheses that might follow the phrase.

In order to implement the data-directed activation of *KS*'s, each *KS* has two components: a pattern and an action. Whenever the pattern is matched by some hypothesis structure on the blackboard, an activation of the *KS* is created. If the *KS* activation is selected eventually by the scheduler, its action is executed in the context of the matched structure. For example, the pattern of a *KS* might be the creation of a new syllable hypothesis and its action might be to use that syllable hypothesis and, possibly, other adjacent syllable hypotheses to create new word hypotheses.

KS activity and hence, the search process, is managed by the scheduler using the focus-of-control database and the scheduling queues. At any point, the *scheduling queues* contain the pending *KS* activations. The scheduler calculates a priority for each waiting activation and selects for execution the one with the highest priority. The priority calculation attempts to estimate the impact of the information to be generated by an activation on the current state of the problem solving. From the problem-solving viewpoint, the impact of some information is a measure of the degree to which it reduces the uncertainty of the interpretation or, alternatively, the degree to which it reduces the number of competing interpretations. This measure changes as the problem solving progresses; thus, the timeliness of creation of the information affects its impact. For example, if two pieces of information can lead to the same hypothesis, the creation of the first of them may have high impact, but the creation of the second will have little, other than adding confirmation to the hypothesis. Lesser *et al.* [16] describe a formal model for this kind of problem-solving activity.

Several dimensions can be used to estimate the impact of information, including the following:

1) The *credibility* of some information is a measure of the system's confidence in the information; the more credible the information, the higher its expected impact.

2) The *scope* of some information is a measure of the amount of the total problem solution that it describes. Scope is related to the level of abstraction (e.g., in speech understanding, a word has larger scope than a syllable) and to the size (e.g., a two-second phrase has larger scope than a one-second phrase). The larger the scope, the greater the impact because a larger portion of the complete interpretation, and hence, more constraint is specified.

3) The *diagnosticity* of some information is a measure of how much competing information can be resolved by the information [12]. For example, if one part of the current partial solution has high credibility while another part has only low credibility, a moderately credible piece of information in the former area will have low diagnosticity, but a moderately credible piece in the latter area will have high diagnosticity and, hence, greater impact.

The *focus-of-control* database contains meta-information about the state of the system's problem-solving activity. The meta-information is used to estimate the impact of information, based on its credibility, scope, and diagnosticity. Meta-information includes such things as the current best hypotheses on the blackboard and how much time has elapsed since these hypotheses were generated or combined with others. (This latter kind of information allows the system to recognize a state of stagnation in part of the problem solving, and then to cause the reappraisal of the impact of the current best hypotheses.) The focus-of-control database is updated by the blackboard monitor based on the generation and modification of hypotheses on the blackboard by *KS*'s.

The blackboard monitor is also used to implement the data-directed activation of *KS*'s. At system initialization, each *KS* declares hypothesis characteristics relevant to it. When an hypothesis is created or modified so as to match those characteristics, the blackboard monitor creates an activation record for the *KS* on that hypothesis and places it in the scheduling queues.

III. Issues in Distributing Hearsay-II

Fig. 2 presents a number of dimensions of decomposition of Hearsay-II for a distributed environment and several options for each dimension. From this table and the overview above, it can be seen that the characteristics of the Hearsay-II organization appear to make it suitable for a distribution along several dimensions.

1) *Information* might be distributed: The blackboard database is multidimensional (with the information levels forming one dimension). Each *KS* activation generally accesses only a small localized subspace within the blackboard.

2) *Processing* might be distributed: Knowledge is encapsulated in *KS* modules that are largely independent, anonymous, and capable of asynchronous execution.

3) *Control* might be distributed: *KS* activation is based on the generation and modification of hypotheses on the blackboard (data-directed control). To the extent that these hypotheses can be distributed, control of *KS* activation can also be distributed. The data-directed form of the hypothesize-and-test paradigm permits *KS*'s to exchange partial results in a cooperative fashion.

Given these possibilities, it would appear that the Hearsay-II organization could be decomposed easily for a distributed environment so as to emulate efficiently and exactly the processing that occurs in the centralized version of the organization. In fact, a shared-memory multiprocessor implementation, using explicit synchronization techniques to maintain data integrity and distributed along the processing and control dimensions, achieved significant parallelism—a speedup factor of six [7]. However, the following characteristics of Hearsay-II introduce a number of difficulties for such a straightforward emulation in a distributed environment:

1) the scheduler, which requires a global view of the pending *KS* instantiations (scheduling queues) and the focus-of-control database, is centralized,

2) the blackboard monitor, which updates the focus-of-control database and scheduling queues when a specific type of blackboard change occurs, is centralized, and

3) the patterns of *KS* access to the blackboard overlap, prohibiting the construction of compartmentalized subspaces of the blackboard accessed exclusively by small groups of *KS*'s.

Because there are many *KS* executions, each accessing the blackboard frequently, an extensive amount of interprocessor communication would be required to emulate exactly a centralized view of the blackboard, scheduling queues, and focus-of-control database. The dynamic information in these data structures controls the degree and nature of *KS* cooperation and is essential to the effective implementation of the hypothesize-and-test problem-solving strategy.

Given that the communication and synchronization costs of emulating perfectly the centralized views are too high, one is led to their approximation. The amount and range of internode communication can be reduced, leading to inconsistency and incompleteness of the local views and thus, unnecessary, redundant, and incorrect processing. Experiments with the shared-memory multiprocessor Hearsay-II speech-understanding system described above demonstrated that the system could operate in such an environment [7]. In these experiments, the explicit synchronization was eliminated without degrading accuracy as measured at the end of processing, with an attendant increase in the speedup factor from 6–15 because of the reduction in interprocess interference.

The explanation for this phenomenon is that the asynchronous, data-directed control can apply knowledge to correct certain types of internal errors. Consider the normal activity sequence of a *KS*, which involves first examining the blackboard and then creating new hypotheses on the basis of the examined hypotheses. If the set of relevant hypotheses changes after the *KS* looks at them and before it modifies the blackboard, the modification is inconsistent or incomplete with respect to the current state of the blackboard; however, because of the data-directed nature of *KS* activation, the intervening changes will trigger the same *KS* to recalculate its modifications and, perhaps, generate new alternative hypotheses that are more consistent and/or complete. In addition, other types of inconsistency can be resolved because a complete solution is pieced together from mutually constraining information; thus, additional *KS* processing will usually produce lower credibility ratings for an incorrect hypothesis and its extensions, lessening the likelihood that these incorrect hypotheses will be considered further. This process occurs whether the incorrect hypothesis resulted from a synchronization error, from a mistake in the knowledge used by the *KS*, or from erroneous data. Thus, this self-correcting nature of information flow among *KS*'s, created through the use of the incremental data-directed hypothesize-and-test paradigm, in many cases obviates the need for explicit use of synchronization.

The key issue is whether a distributed decomposition of a Hearsay-II-like system can be designed that can deal with the errors introduced by the approximate emulation well enough to maintain satisfaction of the sufficiency criteria of Section II-A. In the distributed system, internode communication becomes part of the "computing resources" that must be limited for effective system performance.

** INFORMATION **

Distribution of the blackboard:

The blackboard is distributed across the nodes with no duplication of information.

The blackboard is distributed with possible duplication of information; synchronization techniques are used to insure consistency.

The blackboard is distributed with possible duplications and inconsistencies.

Transmission of hypotheses:

Hypotheses are not transmitted beyond the node in which they are created.

Hypotheses may be transmitted directly to a subset of nodes.

Hypotheses may be transmitted directly to all nodes.

In addition, the transmission and reception of hypotheses can be filtered based on characteristics of the hypotheses, e.g., type of hypothesis (information level), credibility rating, and location of the "event" the hypothesis describes.

** PROCESSING **

Distribution of KS's:

Each node has only one *KS*.

Each node has a subset of *KS*'s. The selection might depend on factors such as the type of sensors at the node, the node's physical location, and the input/output characteristics of the *KS*'s.

Each node has all *KS*'s.

Access to the blackboard by KS's:

A *KS* activation can access only the blackboard in its local node.

A *KS* activation can access blackboards in a subset of nodes.

A *KS* activation can access blackboards in any node in the network.

** CONTROL **

Distribution of KS activation:

A change to an hypothesis activites *KS*'s only within the local node.

A change activates *KS*'s in a subset of nodes.

A change activates *KS*'s in any node.

Distribution of scheduling and focus-of-control:

Each node does its own scheduling, based on local information.

Each subset of nodes has a scheduler.

A single, distributed database is used for scheduling.

Fig. 2. Dimensions of decomposition for Hearsay-II.

IV. A Network of Hearsay-II Systems

A primary goal of our decomposition design is to minimize internode communication relative to intranode processing. Because of this and the relatively fine granularity of *KS* activity within a Hearsay-II system, a node must be able to complete a number of *KS* executions in a self-directed way, i.e., without internode communication. Thus, each node in the network must contain *KS*'s, a scheduler and focus-of-control database for selecting the next *KS* activation to execute at each step, a blackboard for *KS* communication, and a blackboard monitor for *KS* activation. Therefore, each node is an architecturally complete Hearsay-II system.

There are dual points from which to view the distribution of the dynamic information (i.e., partial interpretations and meta-information) in the network:

1) A virtual global database represents all the system's information; the local databases at each node contain the node's partial view of the virtual global database, perhaps with some inconsistencies (because of limited internode communication and synchronization).

2) Each node has its own databases; the union of these across all the nodes, with any inconsistencies, represents the total system interpretation.

The first viewpoint corresponds to the way most distributed computing systems are considered—a centralized system is *decomposed*, with each piece (node) in the decomposition viewed as a part of the whole system. From the second viewpoint, the distributed system is *synthesized* from systems operating at each node. The second approach shifts the view from that of a system distributed over a network to that of a network of cooperating systems, each able to perform significant, local, self-directed processing. Another way of distinguishing these viewpoints is that the first considers each node from the context of the whole system, while the second considers the system from the context of the individual node. When considering any particular design choice, one or the other of these viewpoints might be more appropriate. From either viewpoint, the major design decisions are the selection and focusing of knowledge sources at each node and the choice of mechanisms and policies for internode communication to permit effective cooperative problem solving. We will now describe some possibilities for each of these areas.

A. Intranode Considerations—Selection and Focusing of *KS*'s

Intranode processing can be maximized relative to internode communication if *KS* activity is such that the inputs needed by *KS* actions are available on the node's blackboard. Thus, the selection of *KS*'s for each node and the focusing of their activity on particular portions of the problem greatly affects this goal.

The blackboard in a Hearsay-II system is described along several dimensions. One of these is *information level;* this dimension has discrete points, each corresponding to a different way of representing the situation being interpreted. A *KS* typically works with a small number of information levels by noticing one or more hypotheses (called the "stimulus") at one or two levels and by creating new hypotheses or *modifying* existing ones (the *KS*'s "response") at one or two levels. For a collection of *KS*'s to be connected across levels, then, it must be that any level used by some *KS* as its stimulus is used by some *KS* as its response. There are also *KS*'s that are transducers between the system (i.e., the blackboard) and the external world. For the purposes of this discussion, we will think of an input transducer as having no blackboard stimulus and an output transducer as having no blackboard response. In a network of Hearsay-II systems, if a particular node has a *KS* which is level-disconnected on its stimulus or response side, that node is forced to communicate with other nodes to supply the missing stimulus or to provide a use for the "extra" response. Since a primary goal is to maximize intranode processing relative to internode communication, the selection of *KS*'s for each node should maximize the level connectivity. Likewise, transducer *KS*'s should be selected for their appropriateness to the particular types of sensors (and effectors) at the node.

In addition to the information level, there is an orthogonal dimension (or set of dimensions) for locating hypotheses in the blackboard—this is the *location* of the event which the hypothesis describes. For signal interpretation tasks this usually represents a physical location. In speech understanding, for example, most hypotheses (phones, syllables, words, phrases, etc.) can be located as segments on the dimension of time within the utterance. For image understanding, objects (at any of the levels) can be located in the two or three dimensions of the image space. For radar tracking of aircraft, signals and objects can be located in the three-dimensional world. In general, hypotheses closer in the location dimension are more likely to be relevant to each other and to be needed jointly for further *KS* activity. For example, a word hypothesis is likely to be created from adjacent syllable hypotheses, an object from surfaces near each other, and a signal group from signals detected nearby. Thus, a node should attempt to acquire for its local blackboard all of the hypotheses at a given level within a contiguous segment in the location dimension(s).

All levels in the system taken together with the full extent of the location dimension(s) define a node's largest possible scope. The term *area-of-interest* will be used to denote, for each node, that portion of the maximum scope representable within the node's local blackboard.

The levels in the area-of-interest are the union of the stimulus and response levels of the *KS*'s in the node—any other levels would be useless to the node.[3] A node's area-of-interest at the information level(s) to which the sensory data is transduced should cover in the location dimensions at least the area covered by the node's sensors; otherwise, some of the sensory data would be lost, since the only direct action the transducer *KS* can take is to create hypotheses on the local blackboard about the data.[4] At the other levels, the location segment

[3] In Section IV-D2), we will show one use for representing hypotheses which cannot be processed by local *KS*'s, in particular, for allowing a node to act as a store-and-forward message handler.

[4] Of course, the transducer could use the sensory information to modify hypotheses about adjacent areas, but this would represent the sensory information only indirectly.

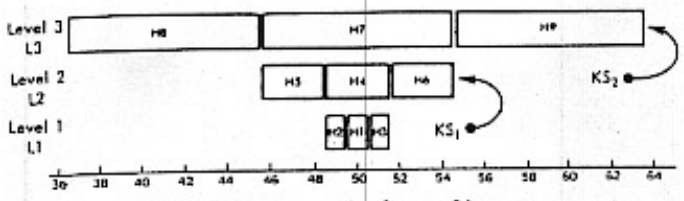

Fig. 3. Simple example of area-of-interest.

should probably include at least the projection of the location segment at the transduction level, since it is reasonable to create higher level hypotheses about the locations covered by the node's sensors. In addition, the location segment should also likely be extended somewhat beyond the range of the local sensors; this is to allow the node to acquire information from neighboring nodes to use as context for *KS* processing. Finally, this context extension should probably be larger at higher information levels because the size of hypotheses [i.e., their length in the location dimension (s)] tend to be larger at the higher levels; e.g., words are usually bigger than syllables, objects are usually bigger than surfaces, and area maps larger than aircraft.

As an aid to understanding the notion of area-of-interest, let us consider a simple example of bottom-up processing at a single node of a network operating in a one-dimensional location space. The node has three information levels, labeled $L1$, $L2$, and $L3$ and two knowledge sources, $KS1$ and $KS2$ (see Fig. 3). Hypotheses on $L1$ are uniformly one unit long in the location dimension and are contiguous and nonoverlapping. The sensor associated with the node produces a single hypothesis on $L1$, called $H1$, at location 50.[5] Knowledge source $KS1$ in the node can take three contiguous hypotheses on $L1$–call them $H2$, $H1$, and $H3$–and produce $H4$ as an abstraction of them on $L2$. Likewise, knowledge source $KS2$ produces hypotheses on $L3$ from triples of hypotheses on $L2$.

In order for $KS1$ to operate, the node must receive hypotheses $H2$ and $H3$ as messages from some other nodes because its local sensor can generate only $H1$. Likewise, for $KS2$ to operate, the $H5$ and $H6$ hypotheses must be received on $L2$. The scope required to be representable on $L2$ is larger than on $L1$. If processing were to continue similarly above $L3$, $L3$'s scope would have to be larger still. Thus, the location dimension of the area-of-interest expands at higher levels. The lateral communication (e.g., $H2$ and $H3$, and $H5$ and $H6$) forms a context for processing and provides a connectivity in the location dimension (*lateral connectivity*), similar to the connectivity in the information-level dimension.

The particular scope of the area-of-interest is dependent on the information required by the *KS*'s. In this simple example, $KS2$ is able to create hypotheses on $L3$ based solely on the information on $L2$. If $KS2$ required information about an $L2$ hypothesis that is not represented in the abstraction on $L2$, it will want to look at the $L1$ substructure of the hypothesis. If the information needed is about $H4$, $KS2$ can access it on the node's blackboard directly, looking at hypotheses $H2$, $H1$, and $H3$. If, however, $KS2$ needs to look at the substructure of $H5$ or $H6$, there is a problem because the $L1$ representations of those hypotheses are not on the node's blackboard. One solution is to have $KS2$ do the best it can without the information, thus requiring no additional internode communication, but introducing additional uncertainty in the problem solving. Another solution to this problem is to extend the node's area-of-interest on $L1$ in order to represent the needed information. This extension can be handled in several ways.

1) *A priori* analysis of $KS2$ indicates that the $L1$ information is likely to be needed. Thus, the scope of the node's area-of-interest on $L1$ is permanently specified to be 46–54, and the node gathers all $L1$ information that it receives. If the needed information is less than the full scope, the expansion of the area can be limited. For example, if information about just boundaries of the $L2$ hypotheses is needed, the scope could be specified as 48–52, rather than 46–54.

2) Each node that transmits $L2$ hypotheses knows that some of the corresponding $L1$ information is likely to be needed; it therefore transmits the relevant $L1$ information whenever it transmits an $L2$ hypothesis. Thus, the scope of the receiving node's area-of-interest on $L1$ dynamically expands in response to the reception of $L2$ hypotheses.

3) When $KS2$ discovers the need for the $L1$ information, it expands the scope of the node's area-of-interest so that it is capable of representing the needed information if it is received. $KS2$ then processes as best it can without the information, perhaps creating no $L3$ hypothesis. If the needed $L1$ information is subsequently received, $KS2$ can be retriggered to reevaluate the earlier action and perform corrective modification if needed.[6]

The suggestions here for defining the area-of-interest of a node are only one possible set of guidelines; others could be used. The area can also be adjusted dynamically to adapt to changing conditions, such as movements of the node or its sensors or changes in demands on the node's processing or memory capacity. What is important is that each node has an area-of-interest that defines its blackboard and thereby puts bounds on the area in which local processing can occur and on what information is important for it to receive. As suggested by the example in this section, the particular sections of the area-of-interest from which information needs to be transmitted and received are task-specific, depending upon the specific requirements of the *KS*'s and their selection and focusing in the network.

B. Network Configurations

Within the guidelines developed so far, a variety of organizational structures can be implemented in the network, depending on the selection and focusing of *KS*'s in each node. For example, if all nodes contain the same set of *KS*'s and levels,

[5] In general, multiple, alternative, competing hypotheses could be produced throughout this example, but we will not consider them here.

[6] There are a variety of approaches for acquiring the needed information which involve more explicit communication among nodes. For example, attached to each transmitted hypothesis is the name of the sender so that later point-to-point communication might be established. Even though the basic approach to internode communication developed here is based on a more implicit communication approach (similar to the way *KS*'s communicate through the blackboard), we briefly discuss some of these more explicit approaches in Section IV-D2).

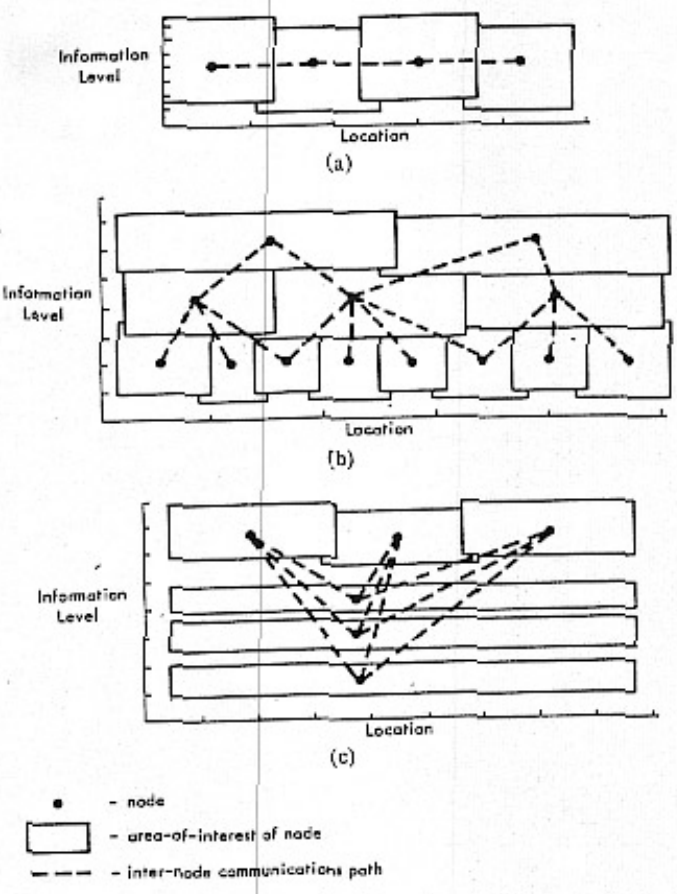

Fig. 4. Schematics of some network configurations. (a) Schematic of a "flat" configuration. (b) Schematic of an overlapping hierarchical configuration. (c) Schematic of a matrix configuration.

the network structure is "flat" and information flow is essentially lateral. This is the simple structure of the system used for the experiments described in the rest of the report. Fig. 4(a) represents such a flat configuration.

More complex processing organizations occur where there is a nonuniform distribution of *KS's* and levels across the nodes. Fig. 4(b) shows an overlapping hierarchical structure. Fig. 4(c) shows the implementation of what is called a "matrix" configuration in organizational structuring (see, for example, [9]). In this configuration, each of a set of general-purpose nodes (at the higher levels) makes use of information from lower level specialists.

Fig. 4 shows simplified schematics of the configurations indicating the levels in each node's area-of-interest, its approximate position in a one-dimensional location scheme, and the internode communication paths. This figure does not indicate the intensity of communication from what sections in an area-of-interest information is being transmitted, whether the paths are bidirectional, or the actual shape of the area-of-interest—varying these parameters leads to greater varieties of network configurations.

The emphasis throughout this report is on the flow of information among nodes, with each node cooperating but having control autonomy. Within this paradigm, various control relationships can be synthesized implicitly by establishing particular information flow paths, resulting in appropriate data-directed activity of nodes. A more explicit implementation of control relationships can be integrated with information flow through the use of a mechanism in Hearsay-II called a *processing goal* [15]. This is an information structure a *KS* creates on the blackboard as an active request for information of a particular type. *KS*'s that can produce such information may then respond to the goal in the same way they would to the creation of a relevant hypothesis. When a goal is transmitted between nodes, as with any other hypothesis, the same kind of request–response activity can occur. A more extended version of this notion, involving a two-way dialogue, is the central idea in the contract net formalism for resource allocation in a distributed environment [21], [22].

C. Internode Communication—Mechanism

In a Hearsay-II system, all inter-*KS* communication is handled indirectly via the creation, modification, and inspection of hypotheses on the blackboard. This same mechanism may be used for internode communication. Consider a Hearsay-II system operating at one node in a network, with its area-of-interest defining the scope of its blackboard and hence the possible areas of attention of its *KS*'s. Now consider adding to that node a transducer *KS* with access to a communication medium (e.g., packet radio) for receiving messages from other nodes describing their hypotheses; if this *RECEIVE KS* modifies its node's blackboard to reflect those messages, other *KS*'s in the node can use this information. Likewise, a *TRANSMIT KS* can select hypotheses on the blackboard and transmit them for reception by other nodes. Fig. 5 shows a network of such systems.

The decision to use the blackboard as the sole means of *KS* interaction in Hearsay-II was made to provide uniformity and to keep *KS*'s relatively independent of each other. The same advantages accrue by using the blackboard for internode communication. A *KS* is triggered by and uses information on the blackboard independent of what other *KS* created it; thus, information placed on the blackboard by the *RECEIVE KS* is automatically usable by the other *KS*'s, indistinguishably from locally generated information. Likewise, each *KS* posts its results on the blackboard without concern for what other *KS*'s might use it; thus, the information to be transmitted by the *TRANSMIT KS* is already available on the blackboard.

A node could transmit, in addition to hypotheses, waiting *KS* activation records from its scheduling queues, in order for them to be executed at another node. If a node receiving such an activation record has both the *KS* and blackboard data needed for executing the activation, the data-directed nature of *KS* activation would have already created an equivalent activation locally. If either the *KS* or data are not present, the activation could not be executed by the receiving node. Thus, it is redundant or useless to share the scheduling queues.[7]

KS's in Hearsay-II interact asynchronously. That is, a *KS*

[7] We are assuming here that the environment for *KS* execution (i.e., the *KS* itself and the relevant blackboard data) is not transmitted. One could consider transmitting such information with *KS* activations for internode load balancing. One could also consider transmitting activations and the node's priority evaluation of them in order to influence the scheduling decisions of other nodes.

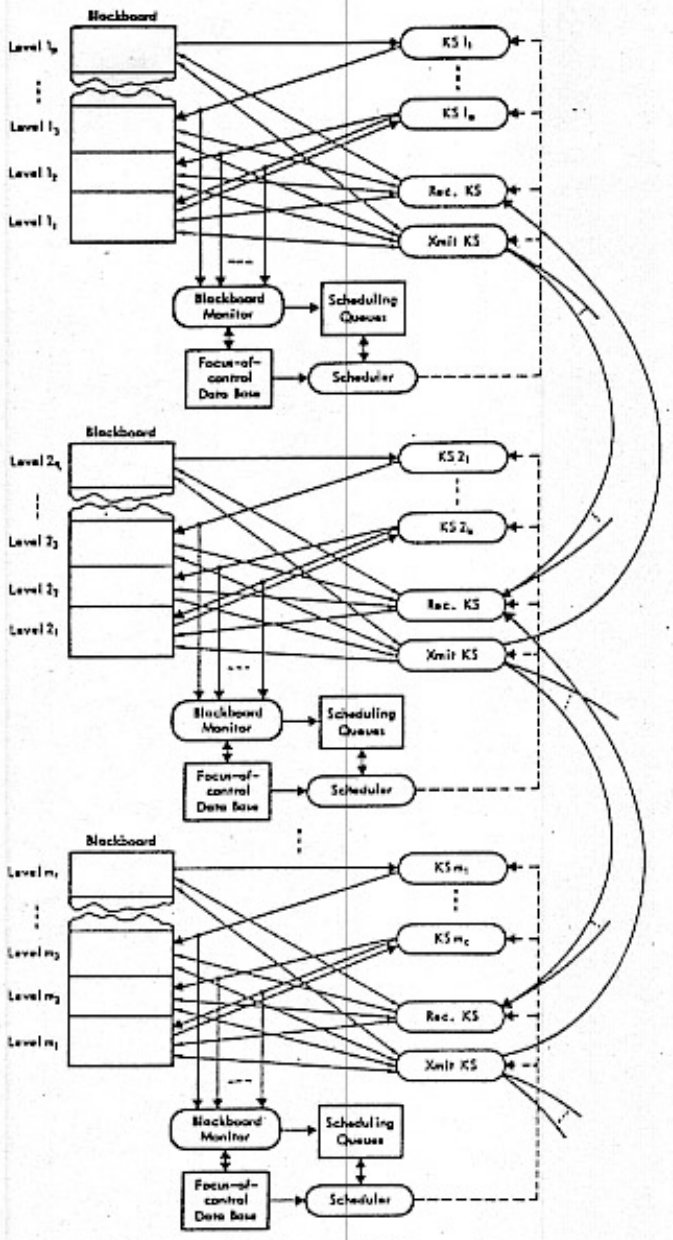

Fig. 5. Schematic of a network of Hearsay-II systems.

triggers whenever an event occurs of interest to it and, when executed, makes use of whatever relevant information is available on the blackboard to make the best statement it can about the situation. Such asynchronous intranode operation naturally allows *KS*'s to handle asynchronous internode communication without modification.

D. Internode Communication—Policies

The ability to run asynchronously eliminates the need for communication costs of synchronization and simplifies the interaction mechanisms. There is still a need to reduce the amount of internode communication while providing each node with the information needed from other nodes (i.e., guaranteeing level and lateral connectivity of *KS* processing). Internode communication can be reduced by limiting the amount of information transmitted, the set of nodes to which any particular message is transmitted, and the distance the message is transmitted.

A centralized Hearsay-II system must limit the number of hypotheses created on its blackboard, in order to avoid a combinatorial explosion of *KS* activity in reaction to these hypotheses. The primary mechanism for limiting the number of hypotheses is the structuring of a *KS* as a generator function. One activation of a *KS* can create a few most credible hypotheses. Stagnation of progress of those hypotheses can trigger new activations to create alternative, less credible hypotheses. Asynchronous *KS* interaction, as described above, permits the additional hypotheses to be exploited in the same manner as the original hypotheses. Similarly, in a distributed system a node need not transmit all its information; rather it can select its "best" and subsequently respond to the need for additional information by transmitting more.

The transmission of a piece of information is worthwhile only if it is received by a node that finds it relevant. At one extreme, each transmission could go to all nodes and each node would be responsible for selecting relevant information from its received communications—this global broadcast scheme would require relatively high bandwidth. Alternatively, the transmitting node could know which other nodes might be interested in the information and, thereby, direct the communication explicitly. The cost of maintaining such a complete distributed knowledge of what is relevant to each node would be high, especially since the information changes as the problem solving progresses. The scheme we consider here is a local transmission based on local knowledge of relevance. Each message is transmitted to a few neighboring nodes. When a node receives information relevant to it, it incorporates the information into its problem-solving state. This action may, in turn, trigger the node to retransmit the information (perhaps modified by its knowledge) on the basis of its local knowledge of relevance.

The transmission of a limited subset of a node's information to a limited subset of other nodes leads to an incremental transmission of information with problem-solving processing at each step, similar to the relaxation paradigm [20]. This transmission scheme results in what can be thought of as a "spreading excitation" of important news through the network. As in relaxation, the propagation of a piece of information dies out as it reaches nodes that find it irrelevant or unimportant.

Local knowledge-based processing at each step of the transmission can serve to correct errors in the information, including errors introduced by the communication process itself. Since communication is incremental, this error correction capability can serve to limit the propagation of errors, as opposed to a global broadcast scheme, which propagates them widely. One drawback of the incremental transmission strategy is the increase in the time needed to communicate important information across the net, because each local step adds some delay. However, a node's information is generally most directly relevant to nodes nearby, and the information contained in these neighboring nodes is generally more constraining (i.e., error-correcting) than that of nodes farther away. Another drawback is the possibility that the transmission of important information will die out because the local measures of importance may be incorrect. This danger is reduced because of the

correlation between the proximity of nodes and their measures of relevance. It can be reduced further by increasing the richness of connectivity of the internode communication paths, at the cost of additional communication.

In order for one node to have information relevant to another, their areas-of-interest must overlap, since each node's area-of-interest defines what is of interest to it. Thus, the selection of areas-of-interest also constrains the potential internode communication patterns. The criteria for selecting the area-of-interest given in Section IV-A led us to place the center of the node's area at the location of the node's sensors. Thus, geographically proximate nodes—i.e., those with sensors proximate in the location dimension—have more overlap in their areas than nodes that are further apart, and therefore have more to communicate.

The incremental communication strategy is also more economical, since communication between nodes is generally less costly the closer they are. This is certainly true if the communication medium is hard-wired lines. It is also true for radio; in fact, as the distance that messages need travel is reduced, the power requirement is reduced (and with it the cost of hardware). Also, the same broadcast channel can be used simultaneously in different parts of the network with less interference.

In order to implement such an incremental communication system, three policies must be specified:

1) the *RECEIVE KS*'s integration of received information onto the blackboard,

2) the *TRANSMIT KS*'s selection of information to transmit, and

3) the determination of which nodes will communicate.

At the heart of these different policies are measures of the relevance (i.e., expected impact) of information for the processing at individual nodes. As described in Section II-B, estimating impact is an important part of the focus-of-control issue for the centralized problem-solving system and meta-information (called the "focus-of-control database") plays a key role in this estimation. Because this meta-information attempts to measure the current state of progress in the problem-solving system, it requires a global view of the problem-solving database (the blackboard). In attempting to develop mechanisms to distribute the meta-information among the nodes, there is s tradeoff between the accuracy and scope of this information on one hand and the cost of acquiring it on the other. The more accurate and globally representative this meta-information, the better the estimate of the relevance of local processing to other nodes. Better estimation leads to lower transmission bandwidth requirements, less redundant processing, and more responsiveness of the system to new, important information. However, the cost of acquiring the more accurate meta-information has its own attendant bandwidth and processing costs that can possibly outweigh the advantages of better local estimates. This tradeoff is classic to all resource-allocation problems, i.e., the cost of doing the allocation (in terms of processing and information acquisition necessary to support it) versus the resources saved by doing it.

1) *The Basic Policy:* The basic policy for communication to be considered is for a node

a) to accept any received information that is representable within its area-of-interest and to integrate that information onto its blackboard as if it were generated by local *KS*'s (and, hence, update its meta-information accordingly),

b) to select for transmission those hypotheses whose estimated impact is highest and that have not been transmitted previously, and

c) to broadcast them to all nodes that can receive the communication directly.

This policy is simple in that communication is not directed to specific receiving nodes, no distinction is made between locally generated and externally received hypotheses, and the mechanism already used to control local activity is also used to select hypotheses to be transmitted.

This policy leads to the same kind of generator behavior that is produced in the local *KS* activity: high-impact hypotheses (locally decided) are transmitted initially. If, after a time, no higher impact hypotheses arrive on the node's blackboard (either generated locally or received from some other node) that subsume or compete with these transmitted hypotheses, the stagnation mechanism will cause other, previously lower rated hypotheses now to be rated high impact and, hence, transmitted.

Since a node's meta-information is strongly dependent on those hypotheses judged high impact, and since it is those hypotheses that are transmitted, a receiving node, by incorporating those hypotheses and modifying its meta-information accordingly, will implicitly incorporate a large part of the sender's relevant meta-information. Thus, the meta-information will also be "relaxed" across the network.

We will now discuss some variants of this basic policy. These respond to particular characteristics of the problem-solving task and the communication channels.

2) *Variants:* If the reliability of the problem-solving processing is such that most hypotheses of small scope are incorrect and if most of the small-scope hypotheses can be refuted by additional processing within the creating node, then it may be better to transmit only hypotheses for which the node has exhausted all of its possible local processing and which come through that processing with a high-impact measure. This strategy, called *locally complete,* can 1) reduce the communication bandwidth needed, since fewer hypotheses need to be sent (just those that survive unrefuted), 2) reduce the processing requirements of the receiving nodes, since they will have fewer hypotheses to incorporate and judge, 3) avoid redundant communication in the case that two nodes have a large area-of-interest overlap, and 4) increase the relevance of transmitted hypotheses because their scopes are larger (due to the additional processing) and are, thus, more likely to overlap areas-of-interest of other nodes. The potential disadvantage is a loss of timeliness—the earlier transmission might provide significant constraint for the receiving node.

A technique we call *murmuring* can be used to improve the reliability of communication. In this technique, a node retransmits high-impact hypotheses. A simple approach is to murmur periodically, independent of other communication. A more efficient approach is to murmur high-impact hypotheses unless the node receives or generates higher impact

hypotheses. The stagnation measures (see Section II-B) can be used to implement this strategy. Murmuring is a knowledge-based technique which can be used to correct for lost communications due to intermittent channel or node failures and to bring up-to-date new or moving nodes, thereby gaining some measure of dynamic network configuration. This mechanism has the advantage of preserving anonymity of communication and requires no explicit handshaking or acknowledgment.

The mechanisms described so far involve the acquisition by each node of a model of the processing state of other nodes implicitly through the problem-solving information received by the node. Such implicit mechanisms are simple, but may not be efficient enough for some cases. For example, the assumption that nodes that can communicate directly have overlapping areas-of-interest is needed to guarantee that relevant and needed information is propagated throughout the network; if, however, there are discontinuities or insufficient redundancy in these overlaps, a more explicit mechanism is needed to guarantee a rich enough connectivity to handle the problem-solving.

One way to handle such problems is for a node to transmit a description of its area-of-interest, explictly indicating what kinds of information it needs and what kinds it can produce, i.e., its *input/output* (*I/O*) *characteristics*. Each node receiving this message responds with a reply containing its I/O characteristics. If the initiating node is unsatisfied with the richness of the neighborhood connectivity implied by the responses, it can transmit another message, indicating which of its I/O requirements are not sufficiently satisfied and requesting its neighbors to ask their neighbors, in turn, to fulfill them. The initiating node can continue expanding the area of its request until all of its requirements are met or until it decides to give up. Subsequently, the intermediate neighbors will act as store-and-forward message processors supporting the desired connectivity. This provides a mechanism for generating explicit communication paths between nodes that have no direct communication capabilities. This may be necessary for some of the more complex network configurations, e.g., as in Fig. 4(c), in which overlapping areas-of-interest do not necessarily imply the geographic proximity of the nodes.

This process can be viewed as the dynamic increase of the area-of-interest of each intermediate node so that it can accept the kind of information it is requested to forward. Even though the intermediate node might do no local problem-solving processing on this information, once it has accepted it, the normal criteria for transmission will handle the forwarding function.

Modification of a node's area-of-interest in response to explicit meta-information can also be used for resource allocation. For example, if a node has completed all possible processing within its area-of-interest and does not expect any new tasks to appear within that area-of-interest for some time, it may be worthwhile for it to advertise for new work, using a mechanism similar to that used for insuring connectivity. Conversely, if a node finds the demands on its local processing power too great, it might shrink its area-of-interest, thereby reducing the domain of its activity. If there is sufficient overlap of areas-of-interest, this action results in just a reduction of redundancy; if the overlap is not sufficient, a renegotiation, using the I/O characteristics, is needed to assure coverage of the whole problem.

It may be useful to transmit other meta-information with hypotheses: for example, the name and location of the sending node, the time the hypothesis was generated, the amount of computing effort expended on the hypothesis, and the number of nodes that previously processed the hypothesis. The receiving node can augment its meta-information with this information.

Fig. 6 summarizes the design decisions we have made along each of the dimensions of Fig. 2.

V. The Experiment

An experiment was performed to determine how the problem-solving behavior of such a network of Hearsay-II systems compares to a centralized system. The aspects of behavior studied include the accuracy of the interpretation, time required, amount of internode communication, and robustness in the face of communication errors. This experiment was a simulation only in part, since it used an actual interpretation system analyzing real data, i.e., the Hearsay-II speech-understanding system [6].

A. Simulating a Network

The simulation aspects of the experiment involved emulating a distributed network of nodes with a broadcast communication structure. This was accomplished by developing a multijob coordination facility for the Decsystem TOPS-10 operating system. This facility coordinates communication and concurrency among a collection of independent jobs, each running a Hearsay-II speech-understanding system. The network communication structure is simulated by a shared file that holds a record of each transmission in the network and additional information, such as when and by which node it was generated and which nodes have read it. All jobs can access this file through an internode communication handler added to the basic Hearsay-II system. The simulation of concurrency among the jobs is accomplished by keeping the jobs' clock-times in step; each time a job makes a request to transmit or receive internode communication, it is suspended if its local processor time is no longer the smallest. In this way, the simulation of concurrency is event-driven rather than sampled; this permits accurate measurement and comparison of concurrent events across simulated nodes.

B. Selection of KS's and Areas-of-Interest

A major design decision in the decomposition of a system is the selection and focusing of *KS* processing at each node. In the case of the distributed Hearsay-II speech-understanding system, the decision was to allocate all the *KS*'s to each node. The area-of-interest for each node has all the information levels, but is restricted to a statically assigned segment of the location dimension, i.e., to a segment of the speech signal. Two aspects of the particular blackboard structure and *KS* configuration of the Hearsay-II system used in this experiment motivate this design.

** INFORMATION **
Distribution of the blackboard:
The scope of a node's local blackboard defines its area-of-interest.
Transmission of hypotheses:
A node transmits hypotheses to a local subset of nodes.

** PROCESSING **
Distribution of KS's:
Each node has a subset of *KS*'s.
Access to the blackboard by KS's:
A KS activation can access only the blackboard in its local node.

** CONTROL **
Distribution of KS activation:
A change to an hypothesis activates *KS*'s only within the local node.
Distribution of scheduling and focus-of-control:
Each node does its own scheduling, based on local information.

Fig. 6. Design decisions for a network of Hearsay-II systems.

The first aspect concerns how hypotheses are located on the blackboard. The information levels of the Hearsay-II speech-understanding system are shown in Fig. 7. The position of an hypothesis on the location dimension is defined by its time segment within the spoken utterance. For example, a hypothesis might be that the word "today" occurred at the word level from ms 100 to ms 600 in the utterance. One can think of each node as having a microphone sensor which acquires its input from a segment of the utterance. As discussed in Section IV-A, it is natural to define a node's area-of-interest as being centered, in the location dimension, over its sensor's area. *Thus we are led to a one-dimensional network with each node listening to some portion of the utterance and with the portions overlapping.*

The second aspect concerns the propagation of information across levels of the blackboard. *KS* processing in this version of the Hearsay-II speech system (see Figs. 7 and 8) is bottom-up and pipelined (without feedback) until the word level is reached; i.e., all segments are created, then all syllables, then a selection of words. Additionally, the context of hypotheses required for *KS*'s operating at these levels is highly localized in terms of position within the utterance—i.e., in the location dimension. Thus, by choosing the areas-of-interest to have sufficient size and overlap in the location dimension, it is possible to guarantee that all bottom-up processing to the word level can be accomplished with no internode communication—i.e., there is no need for communication to maintain lateral connectivity for this processing—at the cost of possible redundant processing. The "sufficient" size and overlap criteria must be such that all possible valid hypotheses at these levels can be hypothesized because their time regions lie totally within at least one node.

Above the word level, the more incremental, data-directed form of processing occurs, in which the context of hypotheses required for *KS* processing cannot be localized in the time dimension. In particular, phrase hypotheses must be transmitted among nodes.

Additionally, *KS* processing at the phrase level often requires the detailed characteristics of the underlying word support for the phrase abstractions. As discussed in the example in Section IV-D.1, there are a number of possible approaches to providing the appropriate information to a node. The approach taken here is to transmit explicitly with each phrase hypothesis the name, rating, and time region characteristics of each word contained in its underlying word support. However, there is still a limitation on the scope of a node's area-of-interest at the phrase level, since local *KS* processing at that level can merge disjoint phrase hypotheses into an enlarged phrase hypothesis only if their juncture at the segment level is contained in the area-of-interest of the node. This requirement must be met in order for the *KS*'s to ascertain that particular acoustic phenomena occur at the juncture. This implies that a received phrase hypothesis should be discarded if it does not overlap the node's area-of-interest at the segment level.

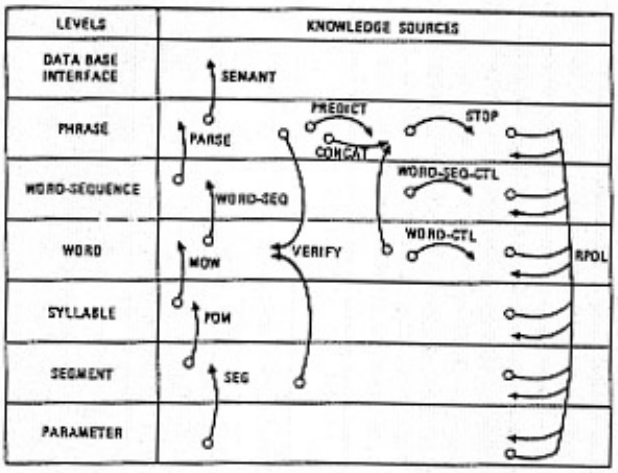

Fig. 7. Levels and knowledge sources of the speech-understanding system. Each *KS* is indicated by one or two vertical arcs, with the circled end indicating the level of its input and the pointed end indicating the level of its output.

C. Communication Strategy

The previous section defines the type of information to transmit (phrase hypotheses and their underlying word support) as well as the policy for its reception (i.e., ignore all received hypotheses that do not overlap the area-of-interest). What remain to be described of the communications strategy are the mechanisms for determining which phrase hypotheses should be transferred and to which nodes they should be sent. Three policies were explored for selecting hypotheses to transmit.

The first policy, called "full transmission," is to have no selection criteria and to transmit each phrase hypothesis as soon as it is created. This policy provides a benchmark for the other policies and simulates a nonsynchronized, centralized blackboard at the phrase level.

The second policy, called "dynamic thresholding," corresponds to the basic policy presented in Section IV-D.1 and uses the local focus-of-control database as a basis for evaluating the importance of a locally generated phrase hypothesis. The focus-of-control database keeps track of the best phrase hypothesis created (or received) for each time area of the utterance. The criterion for "best" hypothesis is constantly reevaluated on the basis of whether a hypothesis has been successfully extended into an enlarged hypothesis—if not, its rating is decreased, possibly resulting in the choice of another hypothesis to replace it as the best in the area. The criterion for transmission using this policy is straightforward: transmit a hypothesis when it becomes the best in its area.

Signal Acquisition, Parameter Extraction, Segmentation, and Labeling:
SEG: Digitizes the signal, measures parameters, and produces a labeled segmentation.
Word Spotting:
POM: Creates syllable-class hypotheses from segments.
MOW: Creates word hypotheses from syllable classes.
WORD-CTL: Controls the number of word hypotheses that MOW creates.
Phrase-Island Generation:
WORD-SEQ: Creates word-sequence hypotheses that represent potential phrases, from word hypotheses and weak grammatical knowledge.
WORD-SEQ-CTL: Control the number of hypotheses that WORD-SEQ creates.
PARSE: Attempts to parse a word-sequence and, if successful, creates a phrase hypothesis from it.
Phrase Extending:
PREDICT: Predicts all possible words that might syntactically precede or follow a given phrase.
VERIFY: Rates the consistency between segment hypotheses and a contiguous word-phrase pair.
CONCAT: Creates a phrase hypothesis from a verified, contiguous word-phrase pair.
Rating, Halting, and Interpretation:
RPOL: Rates the credibility of each new or modified hypothesis, using information placed on the hypothesis by other *KS*'s.
STOP: Decides to halt processing (detects a complete sentence with a sufficiently high rating, or notes the system has exhausted its available resources), and selects the best phrase hypothesis (or a set of complementary phrase hypotheses) as the output.
SEMANT: Generates an unambiguous interpretation for the information-retrieval system which the user has queried.

Fig. 8. The speech-understanding *KS*'s.

The third policy investigated, called "locally complete," is to transmit an hypothesis if there is no more local *KS* processing that can be performed on the hypothesis. This condition is recognized when the acoustic region of an hypothesis "almost" covers the node's acoustic area-of-interest. This policy implements a simplified version of the locally complete strategy presented in Section IV-D.1. This version is simplified since the impact of a locally complete hypothesis is never explicitly evaluated. Rather, the successful extension of a phrase hypothesis to the boundaries of the node's area-of-interest is taken as an implicit indication that the hypothesis is important and should be transmitted. Additionally, in order to minimize the number of hypotheses transmitted, none of the intermediate phrase hypotheses used in the construction of a locally complete hypothesis are transmitted.

Due to the static allocation of the areas-of-interest and the small number of nodes (a maximum of three), a fully connected communication configuration was chosen. Thus, we are not able to test more complicated and selective communication strategies in which a limited subset of nodes receives each transmission. In this broadcast strategy, all nodes receive the message, the sender does not receive a positive acknowledgment that the message has been received correctly, and the receiver does not know the identity of the sender.

VI. Results

There are two main purposes for gathering experimental data on the performance of a network of Hearsay-II systems. The first is to provide empirical evidence for the assertion that the additional uncertainty introduced by distribution can be handled within the basic, uncertainty-resolving mechanisms of the Hearsay-II architecture. The second is to see if there are dynamic interaction phenomena among the nodes that we had not anticipated from our static analysis, particularly phenomena dealing with communication bandwidth and overall performance.

A. Network versus Centralized

The most important experimental results come from comparing the performance of a three-node Hearsay-II system with that of the centralized version. Given the requirements described in Section V-B and the lengths of the utterances in the test data, three nodes is about the maximum that can be used. Both systems were configured with the same task language (called "*S5*"), which has a 250-word vocabulary and a very simple grammar.[8] We chose for test data a set of ten utterances that had been understood correctly by the centralized system.

The nodes in the network were configured with extensive overlap between their areas-of-interest (see Section V-B). Fig. 9 shows the ten sentences and the areas-of-interest for each of them. The locally complete strategy (see Section V-C) was used for internode communication.

The network system correctly understood all ten of the utterances. Thus, the uncertainty introduced by this distribution of the problem solving was handled by the basic Hearsay-II architecture without the need for additional mechanisms. This basic result has been substantiated by consistently correct interpretations in several additional experiments with, in turn, 1) decreased area-of-interest overlaps, 2) less-constraining grammar, 3) alternative communication policies (Section VI-B), and 4) two-node configuration.

Fig. 10 is a summary of the execution costs for running these ten utterances on the network system relative to the costs on the centralized system. The summary is along two dimensions: the processing time and the number of phrase hypotheses generated and transmitted. As described in Section V-B, the selection of areas-of-interest for these experiments has led to a configuration in which all bottom-up processing through the word level can be accomplished with no internode communication. Since the purpose of these experiments is to investigate internode cooperation, as opposed to task-specific parallelism, the times reported are of the processing after that bottom-up phase has completed. Note that the results of the bottom-up phase are used throughout the subsequent processing—in particular, the segment and word hypotheses within a node are constantly used by the node while investigating the extension of phrase hypotheses. The rationale of the distributed design is to avoid the transmission of the word and segment hypotheses to a central site. When reporting processing time in the network case, the time given is the maximum time over the three nodes, which is an estimate of the clock time of the simulated network.

For the network system, three counts of phrase hypotheses are used. First, is the number of phrase hypotheses generated locally by each node, summed over the three nodes. This measures the amount of search more directly than does pro-

[8] The Hearsay-II speech-understanding system is configurable with a varying range of task languages. The use here of a simple language reduced the amount of computing resources required for the experimental runs.

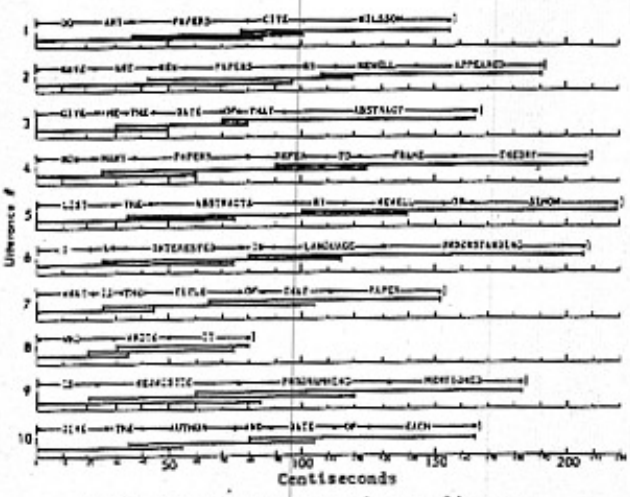

Fig. 9. The test utterances and areas-of-interest.

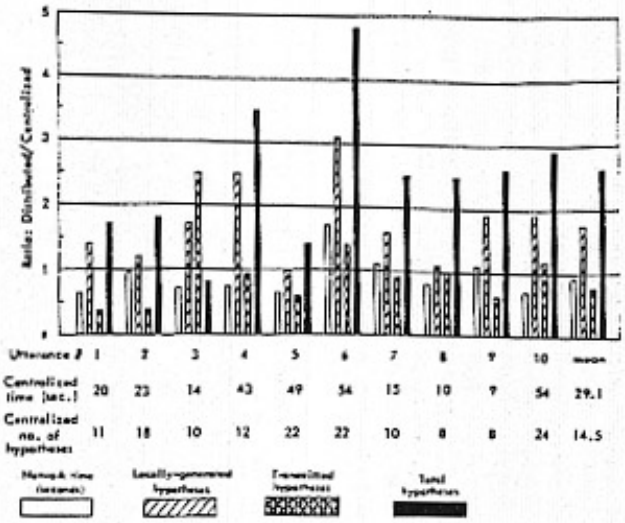

Fig. 10. Performance of the centralized system versus the network system using the locally-complete transmission strategy.

cessing time. Next, is the number of these hypotheses that were selected by the locally complete strategy for **transmission**. This is a measure of the channel costs for communication. Finally, there is the **total number** of phrase hypotheses that occurred; this is the sum over the three nodes of the number of hypotheses created locally by the node and the number of received hypotheses accepted by the node and placed on its blackboard.[9] For each of these three measures, Fig. 10 gives the ratio of that number to the number of hypotheses created in the centralized system.

The major conclusions that can be drawn from the summary statistics in Fig. 10 are as follows:

1) Effective cooperation was achieved among the nodes even though only 44 percent of the locally generated hypotheses were transmitted. This represents 77 percent of the number of hypotheses created in the centralized runs.

2) There was a slight speedup of 10 percent in performing the interpretation above the word level with three nodes. Thus, the interpretation took 2.7 ($=3 \times 0.9$) times as much processing as compared to the centralized version. (Recall that the times reported are of the high-level, highly cooperative processing only. If the bottom-up processing is included, which accounts for about half the time in the centralized system, there is an overall speedup of about 60 percent for the three-node configuration over the centralized version).

We classify the increase in the total amount of high-level processing into three areas: communication, incomplete information for knowledge application, and incomplete meta-information for focusing.

Communication costs include deciding which hypotheses to transmit and accept as well as the physical act of message passing. Also, the receiving node must merge accepted hypotheses into its blackboard structure. These sending and receiving functions account for about six percent of the processing time. To reduce the size of each message, the grammatical structure of the phrase hypothesis is not transmitted; rather, the receiving node recomputes that structure when needed, thus trading off additional processing for reduced communication bandwidth. None of these processing costs occurs in the centralized system.

Incomplete information makes it more costly to process hypotheses. For example, in the centralized system the *PREDICT KS* uses the heuristic of attempting to extend first in the direction with the fewer number of predicted words (i.e., either at the beginning or end of the phrase). In the distributed system, this heuristic often cannot be exploited because the preferred direction would carry the prediction outside the node's area-of-interest. The inability to predict in the direction of greater constraint leads to more word verification processing. A more subtle effect of a node's limited area-of-interest is a shift in the distribution of the length of phrase hypotheses towards hypotheses having fewer words. In general, shorter phrase hypotheses have less grammatical constraint on the number of words they predict, leading to additional word verification. These effects showed up as a doubling of the number of words predicted per phrase hypothesis.

Incomplete meta-information can lead to redundant search and unnecessary search (i.e., with a low likelihood of a correct solution), which reduce the potential speed-up benefits of a parallel search. *Redundant search* occurs because there is no centralized scheduler to coordinate the search of nodes with overlapping areas-of-interest. *Unnecessary search* occurs because the search paradigm is opportunistic across the length of the utterance, i.e., working out from a few islands of reliability discovered in the data. These islands are not, in general, distributed uniformly among the nodes in the network. This leads to cases in which a particular node can do little effective processing until it receives constraining information, i.e., a reliable island, from another node. Likewise, after a node has fully explored all of its reliable islands, it may have little effective processing to do. The processing occurring before the node receives a reliable island and the processing after it has fully exploited all of its reliable islands is, from a global view, unnecessary. Thus, the opportunistic scheduling partially sequentializes the search. The effect this has on the parallel speed-up in a network system depends on the distribution of

[9] This third number may be more or less than the sum of the other two because a transmitted hypothesis is accepted by a receiving node only if it overlaps the node's area-of-interest. Thus, an hypothesis transmitted in a three-node network might be accepted by zero, one, or two nodes.

islands across the nodes—the more uniform the distribution, the greater the speed-up. Fig. 11 illustrates this by showing how one of the test utterances was recognized in the centralized and distributed systems.

Because of the uncertainty in knowledge and data in speech understanding, such unnecessary search may produce hypotheses with sufficient credibility and scope to be transmitted. This internode communication is itself unnecessary and may distract nodes doing productive work, thus causing even more unnecessary search. This distraction occurs because the estimate of impact of an hypothesis is based in part on its scope (length). Thus, a long, moderately rated hypothesis may be considered to have more impact than a short, highly rated one. If a node lacking a reliable island does not soon receive constraining hypotheses, it is often able to develop hypotheses of moderate credibility and large scope which it then transmits. If such an hypothesis is received by a node with a highly reliable island before it has been able to develop that island fully, the node may switch its attention to the longer, received hypothesis, thus delaying, perhaps indefinitely, the useful processing of the shorter, highly credible island. The recognition trace of the utterance shown in Fig. 12 shows the results of such distraction.

This method of estimating impact for focusing decisions is reasonable in a centralized system in which all the input data are received together. In such a system, the development of hypotheses is implicitly more synchronized—the higher rated island would have been extended before the lower rated hypothesis would have been developed. A possible solution to this problem in the network system is to normalize the estimate of impact of received hypotheses according to the scope of the largest locally generated ones.[10]

Five utterances were also run using a more complex (i.e., less constraining) grammar, called "S15." Again, all five were recognized by both the centralized and three-node configurations, adding credence to our hypothesis that the accuracy of the problem solving can be maintained within the distributed configuration. In these runs, the overall speedup increased to 30 percent from the 10 percent of the simpler grammar, indicating more parallelism in the larger search space. The fraction of hypotheses transmitted remained similar to the fraction in the simpler grammar runs.

B. Transmission Policies

The network data in the previous section were generated using the locally complete transmission policy. Fig. 13 presents experimental comparisons of that policy with those of dynamic thresholding and full transmission. (See Section V-C for descriptions of these policies.) The utterances used were the first five of the ten used in the previous section; the same areas-of-interest were used. All five utterances were correctly understood under all three transmission policies.

On the basis of both processing time and number of hypotheses transmitted, locally complete is more efficient than the dynamic thresholding, which in turn is better than the full

[10] It might be desirable to expand such differential treatment of received hypotheses, e.g., to use meta-information about the transmitting node for evaluating the received hypothesis.

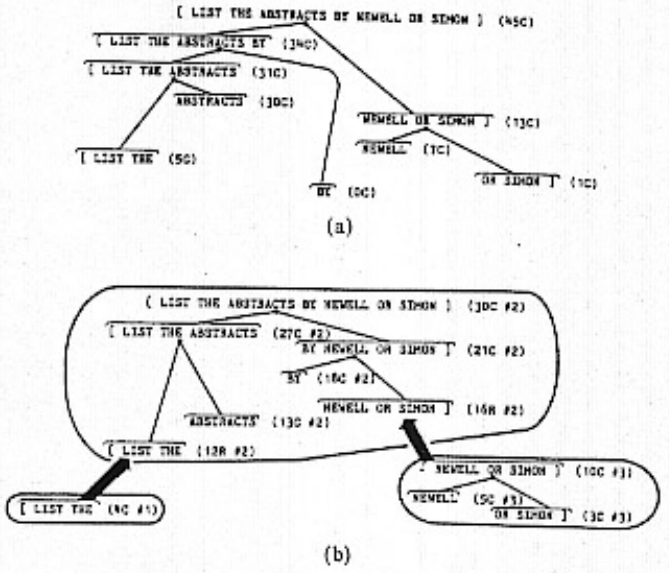

Fig. 11. Recognition process for those partial interpretations of utterance #5 that led to the correct overall interpretation. Joined lines indicate intranode hypothesis creation. Arrows show internode communication of an hypothesis. Numbers in parentheses indicate network processing time in seconds when the hypothesis was created (*C*) or received as a message (*R*). In the multinode case, a second number indicates the node number (e.g., #2 for node two). (a) In the centralized system. (b) In the three-node configuration.

transmission. It thus appears that the timeliness advantage of the dynamic thresholding policy is dominated by the reductions in redundant processing and distracting communication of the locally complete. In some experiments with a more complex grammar, the differential between the two selective policies was reduced—our conjecture is that the extra timeliness of the dynamic thresholding policy becomes more important as the complexity of the search increases.

C. Communication with Errors

In order to assess the robustness of the network system with respect to communication errors, experiments were run in which messages received by a node are randomly discarded with a specified probability. This serves to model communication systems with good error detection but poor correction capabilities, e.g., packet radio. Selection at the receiving end allows for cases in which a broadcast message is received successfully by some nodes but not others.

Two characteristics of the network system should make it robust in the face of communication errors. First, there are redundancies that can recreate the information in lost messages; and second, the system can exploit the recreated information even though it arrives later than would have the original, lost communication. There are several ways of recreating the lost information.

1) The overlapping of areas-of-interest leads to the possibility of creating redundant information directly.

2) The transmission policy can introduce redundant communications. For example, the dynamic threshold policy (and the full transmission policy) can produce a sequence of messages representing the stages of development of a partial solution. Each message in the sequence subsumes the information in the previous messages. This redundancy does not exist in the

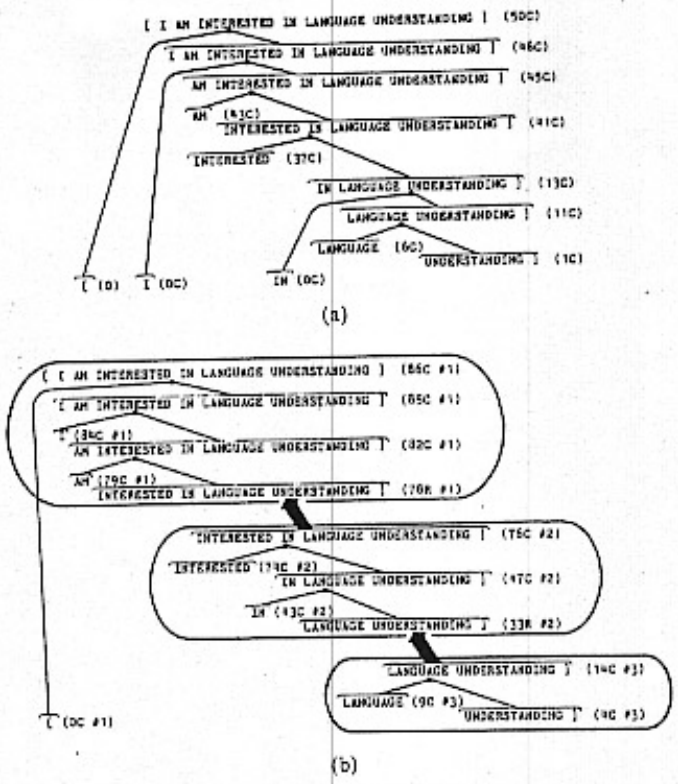

Fig. 12. Recognition process for those partial interpretations of utterance #6 that led to the correct overall interpretation. (a) In the centralized system. (b) In the three-node configuration.

locally complete policy, which transmits only the final message in that sequence; it is for this reason that dynamic threshold was used for this experiment. Other mechanisms, such as murmuring (Section IV-D), can be used for additional explicit redundant communication—we have not explored them in these experiments.

3) The broadcasting of messages makes it possible that messages might be lost to one node, but received by another. The node that correctly receives the information might operate on it and subsequently broadcast a message based on information in the original message. The rebroadcast may be received by the node that lost the original version. This propagation of information among the nodes thus implicitly creates redundant communication paths.

4) The method of building an interpretation by incremental aggregation of partial interpretations makes it possible to derive a correct interpretation in multiple ways. This kind of behavior has been observed in the centralized version of the Hearsay-II speech-understanding system—for example, cases have occurred in which a complete interpretation could not be constructed from one correct island of reliability because of *KS* errors, but could be derived from another. Because a particular message may not be crucial for all ways of deriving a correct interpretation, its loss does not preclude a correct interpretation.

These experiments used the same data as those in Section VI-B. The dynamic threshold transmission policy was used, to provide more redundancy in communication than the locally complete. Fig. 14 shows the performance with 0, 25, 35, and 50 percent of the messages discarded. One utterance of the five was not correctly recognized (i.e., no complete interpretation

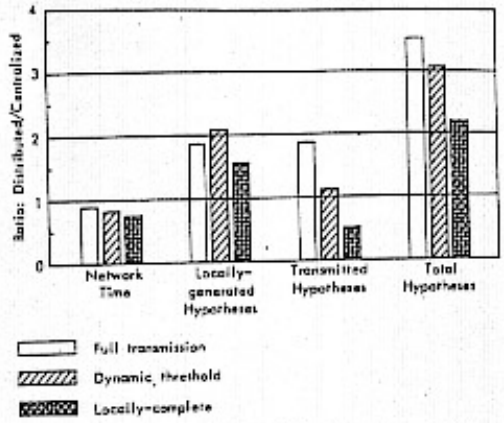

Fig. 13. Performance comparisons of the three transmission policies (means of utterances 1–5).

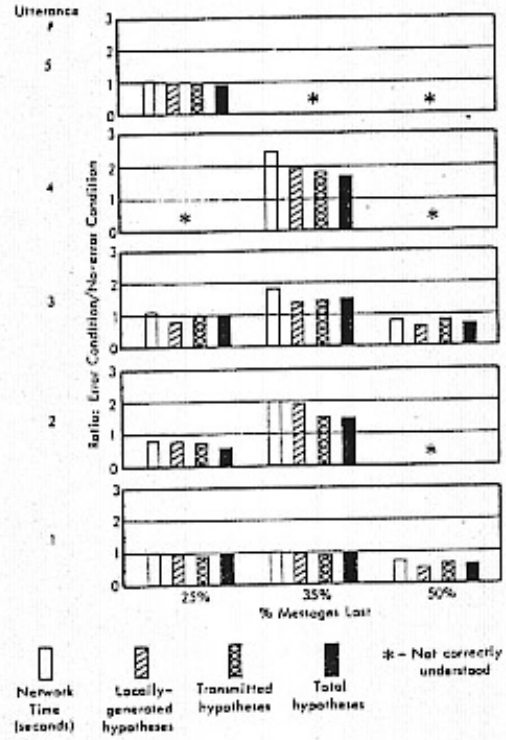

Fig. 14. Performance with communication losses (using the dynamic threshold transmission policy).

was constructed in the maximum allocated processing time) in the 25 and 35 percent cases, and three were missed in the 50 percent case. There are several interesting points about the statistics. For example, the execution times for some runs was decreased *because of* the errorful communication channel. This occurred when messages, discarded due to the simulated communication failures, happened to be either incorrect or redundant. Other runs, as expected, required additional processing time and communication to recreate the nonredundant information lost due to communication failure.

Several runs were not correctly recognized because a message was lost which contained the first or last word in the utterance.[11] Information about these extreme areas is contained in only a single node and is, thus, especially difficult to recreate

[11] Because of the randomness of message lossage, this happened to occur in utterance #2 in the 25 percent error case but not in the 35 percent case.

in another node. The loss of this information is not always fatal. Fig. 15(b) shows an example where first-word information was lost on two separate transmissions ([+HAVE+ANY from node 1 to 2 and [+HAVE+ANY+NEW+PAPERS+BY from node 1 to 3). The system, however, was resilient enough to recreate the information through a roundabout path. Fig. 15(a) is a trace of the system recognizing the utterance when this information was not lost.

In summary, the system's performance with a faulty communication channel lends credence to our belief that the architecture is resilient and permits a tradeoff between the amount of processing and reliability of communication. We further believe that the introduction of a knowledge-based murmuring scheme would correct most of the incorrect runs without increasing communication costs significantly.

VII. Conclusions

Let us review our model for distributed interpretation systems.[12]

1) There is a network of systems (nodes), each of which is able to perform significant local processing in a self-directed way. For example, if a node does not receive a particular piece of information in a given amount of time, it is able to continue processing using whatever information is currently available to it.

2) The parts of the problem a node is responsible for working on is called its *area-of-interest* and is defined by the information it needs and produces. In general, areas-of-interest of the nodes overlap. The local database of a node (i.e., what information it actually has) may be incomplete or inconsistent with respect to the databases of the other nodes. Nodes resolve the uncertainty in their information through an iterative, asynchronous exchange of partial, tentative results at various levels of abstraction.

3) Control of cooperation among the nodes is decentralized and implicit in the autonomous behaviors of the individual nodes. Each node uses its local estimate of the state of problem solving in the network to control its processing (i.e., what new information to generate) and transmissions to other nodes.

This model differs from conventional approaches to distributed system design in its emphasis on dealing with uncertainty and error in control, data, and algorithms caused by the distribution as an integral part of the network problem-solving process. An attractive structure for accomplishing this is an opportunistic problem-solving structure and, in particular, one which has implicit (data-directed) information flow and control flow.

The conventional approach to the design of distributed systems is to overlay some basic, centralized problem-solving strategy with new mechanisms to handle the uncertainty and errors introduced by the distribution. It is our hypothesis that this conventional approach limits both the type of systems that can be distributed effectively and the environments in which

[12] Smith and Davis [23] compare this model with the contract net model. Fox [8] discusses distributed problem-solving models from the viewpoint of organizational theory.

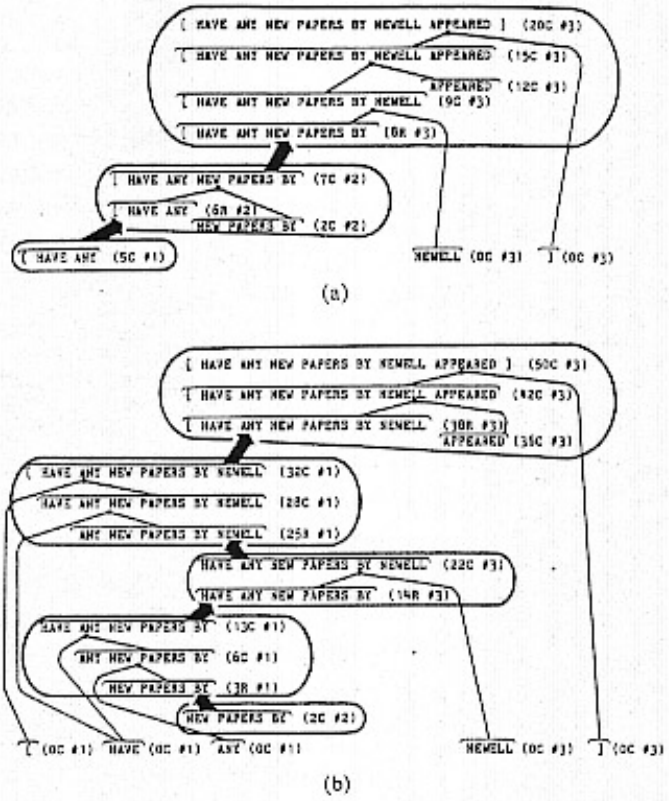

Fig. 15. Trace of utterance #2 processing with and without messages discarded, showing those partial interpretations that led to the correct overall interpretation. (a) With no messages discarded. (b) With 35 percent of the messages discarded.

they can operate. We feel the key to the design of distributed systems is to incorporate mechanisms for dealing with uncertainty and error as an integral part of the problem-solving approach.

The Hearsay-II architecture appears to be a good one for such an integrated approach. The processing can be partitioned or replicated naturally among network nodes because it is already decomposed into independent, self-directed modules (i.e., the *KS*'s), which interact anonymously and are limited in the scope of the data they need and produce. Issues involved in the distribution of the control and data structures of Hearsay-II can be dealt with effectively because of the mechanisms already in the system for resolving uncertainty caused by incomplete or incorrect data and *KS* processing. Let us review these mechanisms and their impact on the ease of system distribution.[13]

Mechanism 1: **Opportunistic nature of information gathering**—Problem-solving is viewed as an incremental, opportunistic, and asynchronous process in which decisions, if they look promising, can be made with incomplete information and later reevaluated in the light of new information.

Impact 1: **Reduced need for synchronization**—Because of this style of problem solving, a node does not have an *a priori* order for processing information and can exploit incomplete

[13] Not all these mechanisms were exploited in the distributed Hearsay-II speech-understanding system described in the previous section. In general, the possibility for exploiting a particular mechanism is dependent on the specifics of the problem-solving application being distributed.

local information. Thus, the processing order within nodes and the transmission of information among nodes does not need to be synchronized.

Mechanism 2: **Use of abstract information**—Because the problem-solving database is structured as a loose hierarchy of increasingly more abstract problem representations, an abstract representation of one aspect of the solution can be used to constrain analysis of other aspects of the problem.

Impact 2: **Reduced internode communication bandwidth**—The ability to use abstract information permits nodes to cooperate by using messages with high information content; thus, the communication bandwidth needed for effective cooperation is reduced.

Mechanism 3: **Incremental aggregation**—A solution is constructed through the incremental piecing together of mutually contraining and consistent information; incorrect partial solutions naturally die out as a result of this process.

Impact 3: **Automatic error detection**—This method of problem solving allows a distributed system to detect and reduce the impact of incorrect decisions caused by incomplete and inconsistent local databases and communication losses.

Mechanism 4: **Problem solving as a search process**—Because of uncertainty in data and *KS* processing, many alternative partial solutions need to be examined in the process of constructing a complete and consistent solution; in this search process, the more uncertainty there exists, the larger the number of alternatives that, in general, have to be explored.

Impact 4: **Internode parallelism**—The requirement that many alternative partial solutions need to be examined generates the possibility that this search can be carried out in parallel by different nodes. The asynchronous nature of information gathering introduces the possibility for additional parallelism, since different aspects of the problem and different information levels can be worked on independently. Further, the introduction of additional uncertainty through incomplete and inconsistent local databases can be traded off against more search—to the degree that this extra search can be done in parallel and does not itself generate proportionally more internode communication, internode bandwidth can be lowered without significant degradation in system response time.

Mechanism 5: **Functionally-accurate definition of solution**—Due to the opportunistic nature of processing and the existence of diverse and overlapping *KS*'s, the correct solution may be derivable in different ways, i.e., using different ordering sequences for incrementally constructing the solution components or using different solution components. Because a solution is based on a set of mutually constraining pieces of information, it is also possible for a correct solution to incorporate information that is correct but not considered very likely, or to use incorrect information that is considered very likely.

Impact 5: **Self-correcting**—Because there are multiple paths from which a solution can be derived, it is possible to correct for what would be considered fatal errors in a conventional distributed problem-solving system. Additionally, system reliability can be varied without modifying the basic problem-solving structure, through the appropriate selection and focusing of local node processing. For example, it is possible to improve reliability by enlarging the overlap among nodes' areas-of-interest, thus increasing the likelihood of generating redundant information. This increases the number of alternative ways that a solution can be derived.

Within the basic distributed problem-solving structure defined by these mechanisms, several other mechanisms have been incorporated or proposed to handle issues specific to a distributed environment.

1) To limit internode communication, an incremental transmission mechanism (with processing at each step) has been developed in which only a limited subset of a node's information is transmitted and to only a limited subset of nodes. A node acts as a generator, which transmits only a few most credible pieces of information and which can subsequently respond to stagnation of progress by producing alternative information. As part of this approach, two policies ("dynamic thresholding" and "locally complete") have been developed for controlling the generator function.

2) To increase network reliability, a knowledge-based mechanism called "murmuring" has been proposed. Here, a node retransmits high-impact information if during a specified time interval it neither receives nor generates higher impact information. Murmuring can be used to correct for lost communications due to intermittent channel or node failures and to bring new or moving nodes up-to-date.

3) To guarantee the appropriate communication connectivity among nodes, a decentralized mechanism for constructing a communication network has been developed. Using this mechanism, which relies on descriptions of the I/O characteristics of each node, nodes act as store-and-forward message processors to provide needed connectivity. A similar mechanism can be used for dynamic allocation of processing tasks among nodes.

4) To provide more sensitive implicit internode control while still retaining decentralization, each node may transmit explicitly its local control information ("meta-information"). Nodes can, thus, determine more directly the state of processing in other nodes.

The experiments described here explore these mechanisms in only a limited way. A number of issues need to be resolved in order to gain an understanding of the more general applicability of this approach.

Distributed Focus of Control:

1) How to coordinate in a decentralized and implicit way the activity of nodes that have overlapping (i.e., redundant) information, so as to control redundant computation, and

2) How to decide locally that a node is performing unnecessary computation and how to select the aspects of the problem on which it should instead focus its attention. This is the problem of dynamic allocation of information and processing capabilities of the network.[14]

Self-correcting Computational Structure:

1) What and how much uncertainty (errors) can be

[14] This issue is related to the classical allocation problem in networks: how to decide if the cost of accessing a distant database is too expensive and whether, instead, the processing should be moved closer to the data or the data moved closer to the processor.

handled using these types of computational structures, and what is the cost in processing and communication to resolve the various types of errors.

Task Characteristics and the Selection of an Appropriate Network Configuration:

1) What characteristics of a task can be used to select a network configuration appropriate for it? When can implicit control and information flow structures be used? Similarly, when should flat, hierarchical, or matrix configurations, or mixtures of them, be used? Candidate characteristics include the patterns of *KS* interaction, the type, spatial distribution, and degree of uncertainty in information, interdependencies of partial interpretations, size of the search space, desired reliability, accuracy, responsiveness and throughput, and available computing resources.

The Hearsay-II speech-understanding system, with only minor changes, performs well as a cooperating network, even though each node has a limited view of the input data. In the experiment with communication losses, system performance degrades gracefully with as much as 50 percent of the messages lost; this experiment also indicates that the system can often compensate automatically for the lost messages by performing additional computation. These results support our general model of distributed systems design. They also indicate that the Hearsay-II architecture is a good one to use as a basis for this approach.

ACKNOWLEDGMENT

The authors wish to thank J. Adams for his help with the programming of the multinode simulation. Also, helpful comments on various drafts of this report were generously given by J. Barnett, D. Corkill, R. Davis, A. Hanson, R. Hayes-Roth, J. Pavlin, D. Schwabe, and Y. Yemini.

REFERENCES

[1] H. G. Barrow and J. M. Tennenbaum, "MSYS: A system for reasoning about scenes," SRI Int., AI Center, Menlo Park, CA, Tech. Rep. 121, Apr. 1976.

[2] G. M. Baudet, "Asynchronous iterative methods for multiprocessors," Comput. Sci. Dep., Carnegie-Mellon Univ., Pittsburgh, PA, Tech. Rep., Nov. 1976.

[3] R. J. Drazovich and S. Brooks, "Surveillance integration automation project (SIAP)," in *Distributed Sensor Nets Workshop*. Pittsburgh, PA: Carnegie-Mellon Univ., Dec. 1978, pp. 119-121.

[4] R. S. Engelmore and H. P. Nii, "A knowledge-based system for the interpretation of protein X-ray crystallographic data," Comput. Sci. Dep. Standard Univ., Stanford, CA, Tech. Rep. Stan-CS-77-589, 1977.

[5] L. D. Erman and V. R. Lesser, "A multilevel organization for problem solving using many diverse cooperating sources of knowledge," in *Proc. 4th Int. Joint Conf. on Artificial Intell.*, Tbilisi, USSR, 1975, pp. 483-490.

[6] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty," *Comput. Surveys*, vol. 12, pp. 213-253, June 1980.

[7] R. D. Fennell and V. R. Lesser, "Parallelism in AI problem-solving: A case study of Hearsay-II," *IEEE Trans. Comput.*, vol. C-26, pp. 98-111, Feb. 1977.

[8] M. S. Fox, "An organizational view of distributed systems," in *Proc. Int. Conf. on Syst. and Cybern.* Denver, CO, Oct. 1979, pp. 354-359.

[9] J. Galbraith, *Designing Complex Organizations*. Addison-Wesley, 1973.

[10] A. R. Hanson and E. M. Riseman, "VISIONS: A computer system for interpreting scenes," in *Computer Vision Systems*, A. Hanson and E. Riseman, Eds. New York: Academic, 1978, pp. 303-333.

[11] F. Hayes-Roth and V. R. Lesser, "Focus of attention in the Hearsay-II speech-understanding system," in *Proc. 5th Int. Joint Conf. on Artificial Intell.*, Cambridge, MA, 1977, pp. 27-35.

[12] F. Hayes-Roth, "The role of partial and best matches in knowledge systems," in *Pattern-Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic, 1978.

[13] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, pp. 1468-1496, Nov. 1978.

[14] V. R. Lesser, R. D. Fennell, L. D. Erman, and D. R. Reddy, "Organization of the Hearsay-II speech understanding system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 23, pp. 11-23, 1975.

[15] V. R. Lesser and L. D. Erman, "A retrospective view of the Hearsay-II architecture," in *Proc. 5th Int. Joint Conf. on Artificial Intell.*, Cambridge, MA, 1977, pp. 790-800.

[16] V. R. Lesser, J. Pavlin, and S. Reed, "Quantifying and simulating the behavior of knowledge-based interpretation systems," in *Proc. 1st Nat. Conf. on Artificial Intell.*, Stanford, CA, Aug. 1980, pp. 111-115.

[17] V. R. Lesser and D. D. Corkill, "Functionally accurate cooperative distributed systems," *IEEE Trans. Syst. Man Cybern.*, 1981, to be published. (This is an expanded version of "Cooperative distributed problem solving: A new approach for structuring distributed systems," Comput. and Inform. Sci. Dep., Univ. of Massachusetts, Tech. Rep. COINS 78-7, May 1978.)

[18] B. T. Lowerre and R. Reddy, "The HARPY speech understanding system," in *Trends in Speech Recognition*, W. A. Lea, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1980, ch. 15.

[19] H. P. Nii and E. A. Feingenbaum, "Rule-based understanding of signals," in *Pattern-Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic, 1978.

[20] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, 1976.

[21] R. G. Smith and R. Davis, "Distributed problem solving: The contract net approach," in *Proc. 2nd Nat. Conf. of Canadian Soc. for Computational Studies of Intell.*, Toronto, Canada, July 1978, pp. 278-287.

[22] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," in *Proc. 1st Int. Conf. on Distributed Comput. Syst.* Huntsville, AL, Oct. 1979, pp. 185-192.

[23] R. G. Smith and R. Davis, "Cooperation in distributed problem solving," in *Proc. Int. Conf. on Syst. and Cybern.*, Denver, CO, Oct. 1979, pp. 366-371.

Victor R. Lesser was born in New York City, on November 21, 1944. He received the A.B. degree in mathematics from Cornell University, Ithaca, NY, in 1966 and the M.S. and Ph.D. degrees in computer science from Stanford University, Stanford, CA, in 1969 and 1972, respectively.

From 1972 to 1977 he was a Research Computer Scientist in the Computer Science Department at Carnegie-Mellon University, Pittsburgh, PA. While there, he was a principal on the Hearsay-II Speech-Understanding Project and was also on the initial design team for the CM* multiprocessor architecture. Since 1977 he has been on the faculty of the Computer and Information Sciences Department of the University of Massachusetts, Amherst, MA where he currently holds the position of Associate Professor. His current research interests focus on system architectures for distributed problem solving and knowledge-based artificial intelligence.

Lee D. Erman was born in Chicago, IL, in 1944. He received the B.S. in mathematics from the University of Michigan, Ann Arbor, in 1966 and the M.S. and Ph.D. degrees in computer science from Stanford University, Stanford, CA, in 1968 and 1974, respectively.

From 1971 to 1978 he was on the faculty of the Carnegie-Mellon University Computer Science Department as Research Associate, Research Computer Scientist, and Assistant Professor. He was a principal designer, implementor, and man-

ager of the Hearsay speech understanding research. Since 1978 he has been on the Research Staff of Information Sciences Institute (ISI) of the University of Southern California, Los Angeles, where he is doing research on converting informal specifications of systems to high-level, formal specifications and on the architectures of systems for implementing knowledge-based systems. His major interests are in knowledge-based artificial intelligence systems in errorful domains, and the tailoring of computer systems for specialized tasks.

Dr. Erman is currently Chairman of the Association for Computing Machinery's. Special Interest Group on Artificial Intelligence (SIGART). He is also on the Editorial Board of the journal *Artificial Intelligence* and the Executive Council of the American Association for Artificial Intelligence.