AD-A012 916

A MULTI-LEVEL ORGANIZATION FOR PROBLEM SOLVING USING MANY, DIVERSE, COOPERATING SOURCES OF KNOWLEDGE

Lee D. Erman, et al

Carnegie-Mellon University

Prepared for:

Air Force Office of Scientific Research
Defense Advanced Research Projects Agency

March 1975

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

A MULTI-LEVEL ORGANIZATION FOR PROBLEM SOLVING USING MANY, DIVERSE, COOPERATING SOURCES OF KNOWLEDGE

Lee D. Erman and Victor R. Lesser

March, 1975

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR - TR - 75 - 1100	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A MULTI-LEVEL ORGANIZATION FOR PROBLEM SOLVING USING MANY, DIVERSE, COOPERATING SOURCES OF KNOWLEDGE		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lee D. Erman, Victor R. Lesser		8. CONTRACT OR GRANT NUMBER(s) F44620-73-C-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University, Dept of Computer Science, Pittsburgh, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101D A02466
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington, VA 22209		12. REPORT DATE March, 1975
		13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Air Force Office of Scientific Research/NM, 1400 Wilson Blvd, Arlington, VA 22209		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An organization is presented for implementing solutions to knowledge-based AI pr[illegible]ems. The hypothesize-and-test paradigm is used as the basis for co[illegible]on among many diverse and independent knowledge sources (KS's). T[illegible] are assumed individually to be errorful and incomplete.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED PRICES SUBJECT TO CHANGE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20/Abstract

A uniform and integrated multi-level structure, the blackboard, holds the current state of the system. Knowledge sources cooperate by creating, accessing, and modifying elements in the blackboard. The activation of a KS is data-driven, based on the occurrence of patterns in the blackboard which match templates specified by the knowledge source.

Each level in the blackboard specifies a different representation of the problem space; the sequence of levels forms a loose hierarchy in which the elements at each level can approximately be described as abstractions of elements at the next lower level. This decompostion can be thought of as an a priori framework of a plan for solving the problem; each level is a generic stage in the plan.

The elements at each level in the blackboard are hypotheses about some aspect of that level. The internal structure of an hypothesis consists of a fixed set of attributes; this set is the same for hypotheses at all levels of representation in the blackboard. These attributes are selected to serve as mechanisms for implementing the data-directed hypothesize-and-test paradigm and for efficient goal-directed scheduling of KS's. Knowledge sources may create networks of structural relationships among hypotheses. These relationships, which are explicit in the blackboard, serve to represent inferences and deductions made by the KS's about the hypotheses; they also allow competing and overlapping partial solutions to be handled in an integrated manner.

The HearsayII speech-understanding system is an implementation of this organization; it is used here as an example for descriptive purposes.

A MULTI-LEVEL ORGANIZATION FOR PROBLEM SOLVING USING MANY, DIVERSE, COOPERATING SOURCES OF KNOWLEDGE

Lee D. Erman and Victor R. Lesser
Computer Science Department[1]
Carnegie-Mellon University
Pittsburgh, Pa. 15213
March, 1975

ABSTRACT

An organization is presented for implementing solutions to knowledge-based AI problems. The hypothesize-and-test paradigm is used as the basis for cooperation among many diverse and independent knowledge sources (KS's). The KS's are assumed individually to be errorful and incomplete.

A uniform and integrated multi-level structure, the blackboard, holds the current state of the system. Knowledge sources cooperate by creating, accessing, and modifying elements in the blackboard. The activation of a KS is data-driven, based on the occurrence of patterns in the blackboard which match templates specified the knowledge source.

Each level in the blackboard specifies a different representation of the problem space; the sequence of levels forms a loose hierarchy in which the elements at each level can approximately be described as abstractions of elements at the next lower level. This decompostion can be thought of as an a priori framework of a plan for solving the problem; each level is a generic stage in the plan.

The elements at each level in the blackboard are hypotheses about some aspect of that level. The internal structure of an hypothesis consists of a fixed set of attributes; this set is the same for hypotheses at all levels of representation in the blackboard. These attributes are selected to serve as mechanisms for implementing the data-directed hypothesize-and-test paradigm and for efficient goal-directed scheduling of KS's. Knowledge sources may create networks of structural relationships among hypotheses. These relationships, which are explicit in the blackboard, serve to represent inferences and deductions made by the KS's about the hypotheses; they also allow competing and overlapping partial solutions to be handled in an integrated manner.

The HearsayII speech-understanding system is an implementation of this organization; it is used here as an example for descriptive purposes.

[1] This research was supported in part by the Defense Advanced Research Projects Agency under contract no. F44620-73-C-0074 and monitored by the Air Force Office of Scientific Research.

INTRODUCTION

This paper describes an organization for knowledge-based artificial intelligence (AI) programs. Although this organization has been derived while developing several generations of speech understanding systems, we feel that it has general application to other domains of large AI problems (e.g., vision,[1] robotics, chess, natural language understanding, and protocol analysis).

Our efforts follow from the early work of Reddy (1966) and Reddy and Vicens (Vicens, 1[illegible]), through the HearsayI system (Reddy, et al., 1973a, 1973b; Erman, 1974), which was the first demonstrable connected-speech understanding system, up through the currently developing HearsayII system (Erman, et al., 1973; Lesser, et al., 1974; Fennell, 1975). These efforts have increasingly focused on the overall system organization for solving the problem; this has resulted in the design and construction of a sophisticated and structured environment within which problem-solving strategies are developed. Others working in this area also consider this aspect important.[2] The HearsayII system will be used here as the primary example for describing the organization.

THE PROBLEM

The class of AI problem that is addressed in this paper is characterized by having a large problem space and the requirement of a large amount of knowledge for its solution. The large amount of explicit knowledge differentiates these problems from other AI areas (e.g., theorem-proving) in which very general "weak" methods are applied using meager amounts of built-in knowledge (Newell, 1969). Further, the knowledge needed covers a wide and diverse set of areas (some examples in the speech understanding problem are signal analysis, acoustic-phonetics, phonology, syntax, semantics, and pragmatics). We call each such area a <u>knowledge-source</u> (KS) and also define a KS to be an agent which embodies the knowledge of its area and which can take actions based on that knowledge.[3]

The sources of knowledge are often incomplete and approximate. This errorful nature may be traced to three sources: First, the <u>theory</u> on which the KS is based may be

---

1 Reddy (1973) is a comparison of the speech and vision problem domains.

2 Newell, et al., (1971) contains an excellent in-depth study of the speech understanding problem. The current state-of-the-art is represented in the papers of the 1974 IEEE Symposium on Speech Recognition (Erman, 1974b; Reddy, 1975). In particular, Barnett (1973, 1975), and Rovner, et al., (1974) also describe highly structured systems; Baker (1974) has a highly structured system based on a simple Markov model.

3 For the purposes of this discussion, a KS can be considered static; i.e., whether a KS learns from experience is an issue that is orthogonal to this organization.

incomplete or incorrect. For example, modern phonological theories, as applied to the speech problem, are often vague and incomplete. Second, the implementation of a KS may be incomplete or incorrect; this may be caused by an incorrect translation of the theory to the program or by an intentionally heuristic implementation of the theory. Finally, the knowledge source may be operating on incorrect or incomplete data supplied to it by other KS's.[1]

As one knowledge source makes errors and creates ambiguities, other KS's must be brought to bear to correct and clarify those actions. This KS cooperation should occur as soon as possible after the introduction of an error or ambiguity in order to limit its ramifications.

A mechanism for providing this high degree of cooperation is the hypothesize-and-test paradigm. In this paradigm, solution-finding is viewed as an iterative process. Each step in the iteration involves a) the creation of an hypothesis, which is an "educated guess" about some aspect of the problem, and b) a test of the plausibility of the hypothesis. Both of these steps use a priori knowledge about the problem, as well as the previously generated hypotheses. This iterative guess-building terminates when a consistent hypothesis is generated which satisfies the requirements of an overall solution.

As a strategy for developing such systems, one needs the ability to add and replace sources of knowledge and to explore different control strategies. Thus, such changes must be relatively easy to accomplish; there must also be ways to evaluate the performance of the system in general and the roles of the various knowledge sources and control strategies in particular. This ability to experiment conveniently with the system is crucial if the amount of knowledge is large and many people are needed to introduce and validate it. One means of helping to provide these flexibilities is to require that KS's be independent.

Because the problems are large and require many computation steps for their solution, the system must be efficient in its computation. This must be certainly true for a "production" application system; however, it must also be reasonably efficient in the development versions because of the experimental way that a complex, knowledge-based system is developed. That is, many iterative runs over a significant amount of test data must be made to develop and evaluate the knowledge sources and control strategies.

---

1 This may also include externally supplied data (e.g., the digitized acoustic wave-form which is the input to the speech-understanding system); the transducers of these data can be considered to be KS's which also introduce error.

MODEL FOR COOPERATION OF KNOWLEDGE SOURCES

The requirement that knowledge sources be independent implies that the functioning (and very existence) of each must not be necessary or crucial to the others. On the other hand, the KS's are required to cooperate in the iterative guess-building, using and correcting one another's guesses; this implies that there must be interaction among the processes. These two opposing requirements have led to a design in which each KS interfaces to the others externally in a uniform way that is identical across KS's and in which no knowledge source knows what or how many other KS's exist. The interface is implemented as a dynamic global data structure, called the blackboard. The primary units in the blackboard are guesses about particular aspects of the problem; these units, which have a uniform structure throughout the blackboard, are called hypotheses. At any time, the blackboard holds the current state of the system; it contains all the guesses about the problem that exist. Subsets of hypotheses represent partial solutions to the entire problem; these may compete with the partial solutions represented by other (perhaps overlapping) subsets.

Each knowledge source may access any information in the blackboard. Each may add information to the blackboard by creating (or deleting) hypotheses, by modifying existing hypotheses, and by establishing or modifying explicit structural relationships among hypotheses. The generation and modification of globally accessible hypotheses is the exclusive means of communication among the diverse KS's. This mechanism of cooperation, which is an implementation of the hypothesize-and-test paradigm, allows a KS to contribute knowledge without being aware of which other KS's will use the information or which KS supplied the information that it used. It is in this way that knowledge sources are made independent and separable. The structural relationships (which are mentioned above and which will be described below) form a network of the hypotheses and are used to represent the deductions and inferences which caused a KS to generate one hypothesis from others. The explicit retention in the blackboard of these dependency relationships is used to hold, among other things, competing hypotheses. Because these are held in an integrated manner, selective backtracking for error recovery and other search strategies can be implemented in an efficient and non-redundant way.

Decomposition of Knowledge

The decomposition of the overall task into various knowledge sources is regarded as being natural; i.e., the units of the decomposition represent those pieces of knowledge which can be distinguished and recognized as being somehow naturally independent.[1] Such a

[1] The approach taken in knowledge source decomposition is not an attempt to characterize

scheme of "inverse decomposition" (or, composition) seems very natural for many problem-solving tasks, and it fits well into the hypothesize-and-test approach to problem-solving. As long as a sufficient "covering set" of knowledge areas required for problem solution is maintained, one can freely add new knowledge sources, or replace or delete old ones. Each knowledge source is self-contained, but each is expected to cooperate with the other knowledge sources that happen to be present in the system at that time.

A knowledge source is specified in three parts: a) the conditions under which it is to be activated (in terms of the conditions in the blackboard in which it is interested), b) the kinds of changes it makes to the blackboard, and 3) a procedural statement (program) of the algorithm which accomplishes those changes. A knowledge source is thus defined as possessing some processing capability which is able to solve some subproblem, given appropriate circumstances for its activation.

Activation of Knowledge Sources

A knowledge source is instantiated as a knowledge-source process whenever the blackboard exhibits characteristics which satisfy a "precondition" of the knowledge source. A <u>precondition</u> of a KS is a description of some partial state of the blackboard which defines when and where the KS can contribute its knowledge by modifying the blackboard. The KS contributes its knowledge through the mechanism of making hypotheses and evaluating and modifying the contributions of other knowledge sources (by verifying and rating or rejecting the hypotheses made by other knowledge sources). A KS carries out these actions with respect to a particular <u>context</u>, the context being some subset of the previously generated hypotheses in the blackboard. Thus, new hypotheses or modifications to existing hypotheses are constructed from the (static) knowledge of the KS and the educated guesses made at some previous time by other knowledge sources.

The modifications made by any given knowledge-source process are expected to trigger further knowledge sources by creating new conditions in the blackboard to which those knowledge sources, in turn, respond. The structure of a hypothesis is so designed as to allow the preconditions of most KS's to be sensitive to a single, simple change in some hypothesis (such as the creation of a new hypothesis of a particular type, a change of a rating, or the creation of a structural link between particular kinds of hypotheses). Through

---

somehow the overall problem solution process and then apply some sort of traffic flow analysis to its internal workings in order to decompose the total process into minimally interacting knowledge sources. Rather, knowledge sources are defined by starting with some intuitive notion about the various pieces of knowledge which could be incorporated in a useful way to help achieve a solution.

this data-directed interpretation of the hypothesize-and-test paradigm, KS's can also exhibit a high degree of asynchronous activity and potential parallelism.[1]

Control schemes in which one KS explicitly invokes other KS's are not appropriate because of the requirement that KS's be independent and because the invocation of a KS may depend on a complex set of conditions which is created by the combined actions of several KS's. Further, such direct-calling schemes complicate KS's by requiring that they contain information about the KS's that they will call. These same arguments apply against a centralized control scheme which is explicitly predefined for a set of KS's.

Decomposition of the Blackboard

The blackboard is partitioned into distinct information levels; each level is used to hold a different representation of the problem space. (Examples of levels in the speech problem are "syntactic", "lexical", "phonetic", and "acoustic"; examples in scene analysis are "picture point", "line segment", "region", and "object".) Associated with each level is a set of primitive elements appropriate for representing the problem at that level. (In the speech system, for example, the elements at the lexical level are the words of the vocabulary to be recognized, while the elements at the phonetic level are the phones (sounds) of English.) Each hypothesis exists at a particular level and is labeled as being a particular element of the set of primitive elements at that level.

The decomposition of the problem space into levels is a natural parallel to the decomposition into KS's of the knowledge that is to be brought to bear. For many KS's, the KS needs to deal with only one or a few levels to apply its knowledge; it need not even be aware of the existence of other levels. Thus, each KS can be made as simple as its knowledge allows; its interface to the rest of the system is in units and concepts which are natural to it. Also, new levels can be added as new sources of knowledge are designed which need to use them. Finally, it will be shown that the multi-level representation allows for efficiently sequencing the activity of the KS's in a non-deterministic manner and for making use of multiprocessing.

[1] One might think of this model for data-directed activation of KS's as a production system (Newell, 1973) which is executed asynchronously. The preconditions correspond to the left-hand sides (conditions) of productions, and the knowledge sources correspond to the right-hand sides (actions) of the productions. Conceptually, these left-hand sides are evaluated continuously. When a precondition is satisfied, an instantiation of the corresponding right-hand side of its production is created; this instantiation is executed at some arbitrary subsequent time (perhaps subject to instantiation scheduling constraints). It is interesting to note that this generalized form of hypothesize-and-test leads to a system organization with some characteristics also similar to QA4 (Rulifson, et al., 1973) and PLANNER (Hewitt, 1972). In particular, there are strong similarities in the data-directed sequencing of processes.

The sequence of levels forms a loose hierarchical structure in which the elements at each level can approximately be described as abstractions of elements at the next lower level.[1] (For example, an utterance is composed of phrases, which are made of words, put together as syllables, each of which can be described as a sequence of phones, each of which is composed of acoustic segments, each of which can be described by a sequence of ten-millisecond intervals with certain kinds of acoustic characteristics.)

Most of the relationships of a hypothesis are with hypotheses at its level or adjacent levels; further, these relationships can usually be derived (by a KS appropriate to the level) without having to delve below the level of abstraction of the hypothesis. This locality of context simplifies the function of knowledge sources. (Or from the other point of view, the decomposition of knowledge into sufficiently simple-acting KS's also simplifies and localizes relationships in the blackboard.)[2]

The decomposition of the blackboard into distinct levels of representation can also be thought of as an a priori framework of a plan for problem-solving. Each level is a generic stage in the plan. The goal at each level is to create and validate hypotheses at that level. The overall goal of the system is to create the most plausible network of hypotheses that sufficiently covers the levels. ('Plausible' and 'sufficiently' here mean "plausible and sufficient in the judgment of the knowledge sources".) In speech understanding, for example, the goal at the phonetic level is a phonetic transcription of the utterance, while the overall goal is a network which connects hypotheses directly derived from the acoustic input to hypotheses which describe the semantic content of the utterance.

The creation or modification of an hypothesis which is based on a context of hypotheses at a lower level (or levels) can be considered an action of synthesis, or abstraction; conversely, manipulations of an hypothesis based on a higher level context can be considered analysis, or elaboration. In order to overcome the errorfulness of the KS's and also make use of their redundant nature, both kinds of action are desirable in the system.[3]

---

1 Many of the ideas here fit neatly into Simon's description of a "nearly decomposable hierarchical system" (Simon, 1962).

2 This simplification of form and interaction is an expected characteristic of a nearly decomposable hierarchical system (ibid.).

3 The use of the terms 'analysis' and 'synthesis' here are reversed from their usual uses in the speech recognition domain. Traditionally, 'synthesis' means going from a higher-level representation (e.g., lexical) to the speech signal, while analysis refers to the other direction. In speech recognition, however, the object is really to synthesize a meaning for the utterance from the pieces of data which make up the speech signal.

Often, the context for an analysis or synthesis action is localized to the level just above or below the level at which the action takes place. However, this is not a requirement; in fact, an action which skips over several levels can serve strongly to direct the activity of the system and thereby significantly prune the search space. Such a jump over levels is equivalent to constructing a major step in a plan. Further, there is no requirement that a jump necessarily be filled in completely (or even partially) if KS's are confident enough in the consistency of the larger step. Thus, the KS's can dynamically define the granularity in the hypothesis network necessary to assure the desired degree of consistency; this granularity may vary at different places in the blackboard, depending on the particular structures that occur.

Appendix A contains a description of the blackboard and KS decompositions for the HearsayII speech-understanding system.

Hypotheses: Structure and Interrelationships

The internal structure of an hypothesis consists of a fixed set of attributes (named fields); this set is the same for hypotheses at all levels of representation in the blackboard. These attributes are selected to serve as mechanisms for implementing the data-directed hypothesize-and-test paradigm.[1] The values of the attributes are defined and modified by the KS's.

Attributes can be grouped into several classes:

The first class of attributes names the hypothesis: it contains the unique name of the hypothesis, the name of its level, and its label from the element set at that level.

The next class of attributes is composed of parameters which rate the hypothesis. These include separate numerical ratings derived from a) a priori information about the hypothesis, b) analysis actions performed on the hypothesis, c) synthesis actions, and d) combinations of (a), (b), and (c).

Another set of attributes contains information about KS attention to the hypothesis. These include a cumulative measure of the amount of computation that has already been expended on the hypothesis as well as suggestions for how much more processing should occur and of what type (e.g., analysis or synthesis).

One very important set of attributes describes the structural relationships with other hypotheses, as described below.

For each problem domain, it is likely that there are other attributes which are basic

[1] In HearsayII, a KS can specify particular attributes of hypotheses at particular levels which it wants to have monitored. Whenever a change is made to one of these monitored attributes, the KS can be activated and notified of the nature of the change. The section below on "Data-Directed Activation of Knowledge Sources" contains a more complete description of this process.

to the problem and which should be provided in the structure of the hypotheses; these form a problem-specific class of attributes. In speech understanding, for instance, time is a fundamental concept, so the HearsayII system has a class of attributes for describing the begin- and end-time and the duration of the event which the hypothesis represents. (These attributes include ways of explicitly representing fuzzy notions of the times.) For vision, likely attributes would include the location and dimension of the element and trajectory information for moving objects.

The capability for arbitrary KS-specific attributes is also included. This can be used by a KS to hold arbitrary information about the hypothesis; in this way a KS need not hold state information about the hypothesis across activations of the KS and allows, for example, the easy implementation of generator functions. If several KS's share knowledge of the name of one of these attributes, each of them can access and modify the attribute's value and thus communicate just as if it were a "standard" attribute; this can be used as an escape mechanism for explicit KS intercommunication.

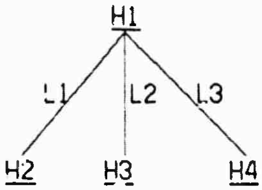
A unique class of hypothesis attributes, called processing state attributes, contains succinct summaries and classifications of the values of the other attributes. For example, the values of the rating attributes are summarized and the hypothesis is classified as either "unrated", "neutral" (noncommittal), "verified", "guaranteed" (strongly verified and unique), or "rejected". Other processing state attributes summarize the structural relationships with other hypotheses and characterize, for example, whether the hypothesis has been "sufficiently and consistently" described synthetically (i.e., as an abstraction of hypotheses at lower levels). The processing state attributes are especially useful for efficiently triggering knowledge sources; for example, a KS may specify in its precondition that it is to be activated whenever a hypothesis at a particular level becomes "verified". These attributes are also used for the goal-directed scheduling of knowledge sources, as described in the next section.

Given a specific hypothesis, a KS can examine the value of any of its attributes. A knowledge source also needs the ability to retrieve sets of hypotheses whose attributes satisfy conditions in which the KS is interested. (E.g., a KS in the speech system may want to find all hypotheses at the phonetic level which are vowels and which occur within a particular time range.) The system provides an associative retrieval search mechanism for accomplishing this. The search condition is specified by a matching-prototype, which is a partial specification of the components of a hypothesis. This partial specification permits a component to be characterized by: a) a set of desired values or b) a don't-care condition. A matching-prototype is applied to a set of hypotheses;[1] those hypotheses whose component values match those specified by the matching-prototype are returned as the result of the search. (Associative retrieval of structural relationships among hypotheses is also provided.) More complex retrievals can be accomplished by combining the retrieval primitives in appropriate ways.

[1] This set can be derived by the KS from several sources. The HearsayII implementation includes the following primitive sources: a) all hypotheses (in the blackboard), b) all hypotheses at a particular level, c) all hypotheses at a particular level whose time attributes overlap a given interval (this provides an extremely efficient, two-dimension partition of the blackboard), and d) all hypotheses whose attributes which are being monitored (for the KS) have changed.

Structural relationships between nodes (hypotheses) in the blackboard are represented through the use of links; links provide a means of specifying contextual abstractions about the relationships of hypotheses. A link is an element which associates two hypotheses as an ordered pair; one of the nodes is termed the upper hypothesis, and the other is called the lower hypothesis. The lower hypothesis is said to support the upper hypothesis while the upper hypothesis is called a use of the lower ·· n general, the lower hypothesis is at the same or a lower level in the blackboard than the upper hypothesis.

There are several types of links, with the types describing various kinds of relationships.[1] Consider this structure:



H1 is the upper hypothesis and H2, H3, and H4 are the lower hypotheses of links L1, L2, and L3, respectively. If the links are all of type OR, the interpretation is that H1 is either an H2 or an H3 or an H4. This is one way that alternative descriptions are possible. If the links in the figure are of type AND, the interpretation is that all of the lower hypotheses are necessary to support the existence of H1. (Note that, in general, all of the supporting (lower) links of a hypothesis are of the same type; one can thus talk of the "type of the hypothesis", which is the same as the type of all of its lower links.)

These two types of node represent different kinds of abstractions: the OR-node specifies a set/member relationship while the AND-node defines a composition abstraction. Variants of the AND- and OR-links are also possible. For example, a SEQUENCE link is similar to the AND-link except that an ordering is implied on the set of lower hypotheses supporting the upper hypothesis. (For the HearsayII speech understanding system, this ordering usually is interpreted as indicating a time ordering of the lower hypotheses.)

Besides showing analysis and synthesis relationships between hypotheses (e.g., that one hypothesis is composed of several other units), a link is a statement about the degree to which one hypothesis implies (i.e., "gives evidence for the existence of") another hypothesis. The strength of the implication is held as attributes of the link. The sense of the implication may be negative; that is, a link may indicate that one hypothesis is evidence for the invalidity of another. This statement of implication may be bi-directional; the existence of the upper hypothesis may give credence to the existence of the lower hypothesis and **vice versa**.

[1] The particular kinds of relationships described here are some of those that were were designed for the speech problem. Although they undoubtedly are not the complete set for all conceivable needs, they do represent the kinds of relationships that need to be and are expressable in the blackboard.

Finally, these relationships can be constructed in an iterative manner; links can be added between existing hypotheses by KS's as they discover new evidence for support.

Just as an hypothesis can have more than one lower link, so it can have several upper links. Each of these represents a different use of the hypothesis; the uses may be competing or complementary. The ability to have multiple uses and supports of the same hypothesis, as opposed to creating duplicates for each competing use and abstraction, serves to keep the blackboard compact and thereby reduces the combinatoric explosion in the search space. Further, since all the information about the hypothesis is localized, all uses and supports of the hypothesis automatically and immediately share any new information added to the hypothesis by any knowledge sources.

A problem with this localization can occur if the interactions between hypotheses span more than one level.[1] In this case, a particular support of the hypothesis (at a lower level) may be inconsistent with one (or more) of the uses of the hypothesis (at a higher level) but is consistent with other uses (or potential uses) of the hypothesis. In order to avoid duplicating the hypothesis, a mechanism called a connection matrix, exists in the system. A connection matrix is an attribute of a hypothesis; its value specifies which of the alternative supports of the hypothesis are applicable ("connected to") which of its uses. The use of a connection matrix allows the results of previous decisions of KS's to be accumulated for future use and modification without necessitating contextual duplication of parts of the data base. This kind of reusage and multiple usage of blackboard structures reduces much of the expensive backtracking that characterizes many problem-solving systems.

Appendix B contains an example of a structure built in the blackboard of the HearsayII system.

Goal-Directed Scheduling of Knowledge Sources

As described earlier, the overall goal of the system is to create the most plausible network of hypotheses that sufficiently spans the levels. At any instant of time, the blackboard may contain many incomplete networks, each of which is plausible as far as it goes. Some of these incomplete networks may also share subnetworks. Through the results of analysis and synthesis actions of knowledge sources, incomplete networks can be expanded (or contracted) and may be joined together (or fragmented). At any time, there may be many places in the blackboard which satisfy the (precondition) contexts for the activation of particular KS's. The task of goal-directed scheduling is to decide to which of these sites to allocate computing resources.

[1] Again, this fits well into Simon's formulation of hierarchical systems.

Several of the attribute classes of a hypothesis can be helpful in making scheduling decisions. Particularly valuable are the values of the attention attributes, which, as described earlier, are indicators telling how much computation has been expended on the hypotheses and suggestions by KS's of how desirable it is to devote further effort on the hypothesis (along with the kinds of processing that are desirable). The processing state attributes are also valuable for making scheduling decisions.

Using these kinds of information, a knowledge source might be scheduled for execution because it possesses the only processing capability available to be applied to an important incompletely explored area of the blackboard. For example, if the blackboard contains focusing factors which highlight activity in a blackboard region in which there are no structural connections between two adjoining levels, the scheduler should give a higher priority to a knowledge source which will attempt (as indicated in its external specifications) to make such a connection than to a knowledge source which is likely merely to perform a minor refinement on the ratings in one of the levels. However, if there are no such processes ready to execute, the scheduling algorithm can perform a type of means-ends analysis in which it schedules those knowledge sources which are likely to produce blackboard changes which, in turn, might trigger the activation of KS's in which the system is currently interested.

The implementation of the goal-directed scheduling strategy is separated from the actions of individual knowledge sources. That is, the decision of whether a KS can contribute in a particular context is local to the KS, while the assignment of that KS to one of the many contexts on which it can possibly operate is made more globally. The three aspects of a) decoupling of focusing strategy from knowledge-source activity, b) decoupling of the data environment (blackboard) from the control flow (KS activation), and c) the limited context in which a KS operates, together permit a quick refocusing of attention of KS's. The ability to refocus quickly is very important because the errorful nature of the KS activity leads to many incomplete and possibly contradictory hypothesis networks; thus, as soon as possible after a network no longer seems promising, the resources of the system should be employed elsewhere.[1]

---

[1] Hayes-Roth et al (1975) describe the implementation of goal-directed scheduling in the HearsayII system.

IMPLEMENTATION OF DATA-DIRECTED ACTIVATION OF KNOWLEDGE SOURCES

Associated with every knowledge source is a specification of the blackboard conditions required for the activation of that knowledge source. This specification, called a precondition, is a decision procedure whose tests are matching-prototypes and structural relationships which, when applied to the blackboard in an associative manner, detect the regions of the blackboard in which the knowledge source is interested. This procedure may contain arbitrarily complex decisions (based on current and past modifications to the blackboard) resulting in the activation of desired knowledge sources within the chosen contexts. The context corresponding to the discovered blackboard region which satisfies some knowledge source's precondition is used as an initial context in which to activate that knowledge source. The efficiency of the KS precondition evaluation is an important aspect of the system's implementation, especially as the knowledge is decomposed into more and smaller KS's and each KS activation requires less computation.

The HearsayII system, as an example of an implementation, makes precondition evaluation efficient by placing additional functions in the routines which modify the blackboard. These functions are activated whenever any KS modifies an attribute in the blackboard which some other KS has asked to be monitored. The essence of the modification is preserved in a data structure, called a change set, which is specific to the attribute changed and the KS which requested the monitoring. A KS specifies in a non-procedural way (either statically or dynamically) those attributes which it wants to monitor. In order to increase the efficiency, monitoring can further be localized to particular levels or even individual hypotheses.

Change sets serve to categorize blackboard modifications (events) and are thus useful in precondition evaluation since they limit the areas in the blackboard that need be examined in detail. As currently implemented in HearsayII, the precondition evaluator of each knowledge source exists as a separate process which monitors changes in the data base (i.e., it monitors additions to those change sets in which the KS is interested). The precondition process is itself data-directed in that it is activated only when sufficient changes have been made in the blackboard (i.e., when an entry is made into one of its change sets, as a side-effect of a relevant blackboard modification). In effect, the precondition processes themselves have preconditions, albeit of a much simpler form than those possible for knowledge sources. For example, a precondition process in the speech system may specify that it should be activated whenever changes occur to two adjacent hypotheses at the word level or whenever support is added to the phrasal level. By using the (coarse) classifications afforded by change sets, the system avoids most unnecessary executions of the precondition processes. The major point is that the scheme of precondition evaluation is event-driven, being based on the occurrence of changes in the blackboard; i.e., it is only at points of modification to the blackboard that a precondition that was previously unsatisfied may become satisfied. In particular, precondition evaluators are not involved in a form of busy waiting in which they are constantly looking for something that is not yet there.

Once invoked, a precondition procedure uses sequences of associative retrievals and structural matches on portions of the blackboard in an attempt to establish a context satisfying the preconditions of one or more of "its" knowledge sources; any given precondition procedure may be responsible for instantiating several (related) knowledge sources. Notice that the data-directed nature of precondition evaluation and knowledge-source activation is linked closely to the primitive functions that are able to modify the data base, for it is only at points of modification that a precondition that was unsatisfied before may become satisfied. Hence, data base modification routines have the responsibility (although perhaps indirectly) of activating the precondition evaluation mechanism.

Implementation on Parallel Computers

Because of the independence of KS's and their data-directed activation, there is a great deal of potential parallelism in this organization. Trends in computer architecture indicate that large amounts of computing power will be economically realized in asynchronous multiprocessor networks. Thus, the implementation of such large AI programs on multiprocessors becomes an attractive goal. There are, however, a set of issues in such an implementation; most of these deal with interference among KS's when they attempt simultaneously to access the blackboard. Effective solutions to these problems have been developed in the HearsayII implementation; Lesser, et al., (1974), Lesser (1975), and Fennell and Lesser (1975) describe these solutions.

ACKNOWLEDGMENTS

We wish to acknowledge the contributions of the following people: Richard Fennell for his major role in the development of HearsayI and II (and, in particular, the multiprocessing aspects of the system organization) and for helpful suggestions for this paper, Raj Reddy for many of the basic ideas which have led to the organization described here, Allen Newell for his insights into the task of problem-solving, Gregory Gill for his untiring efforts in system implementation, and Frederick Hayes-Roth for suggestions for this paper.

Appendix A:
EXAMPLE OF BLACKBOARD AND KS DECOMPOSITION IN HEARSAY II[1]

Figure 1 shows a schematic of the levels of HearsayII.

Conceptual ________
Phrasal ________
Lexical ________
Syllabic ________
Surface-phonemic ________
Phonetic ________
Segmental ________
Parametric ________

Figure 1. The Levels in HearsayII.

<u>Parametric Level</u> - The parametric level holds the most basic representation of the utterance that the system has; it is the only direct input to the machine about the acoustic signal. Several different sets of parameters are being used in HearsayII interchangeably: 1/3-octave filter-band energies measured every 10 msec., LPC-derived vocal-tract parameters, and wide-band energies and zero-crossing counts.

<u>Segmental Level</u> - This level represents the utterance as labeled acoustic segments. Although the set of labels may be phonetic-like, the level is not intended to be phonetic -- the segmentation and labeling reflect acoustic manifestation and do not, for example, attempt to compensate for the context of the segments or attempt to combine acoustically dissimilar segments into (phonetic) units. As with all levels, any particular portion of the utterance may be represented by more than one competing hypothesis (i.e., multiple segmentations and labelings may co-exist).

<u>Phonetic Level</u> - At this level, the utterance is represented by a phonetic description. This is a <u>broad</u> phonetic description in that the size (duration) of the units is on the order of the "size" of phonemes; it is a <u>fine</u> phonetic description to the extent that each element is labeled with a fairly detailed allophonic classification (e.g., "stressed, nasalized [I]").

<u>Surface-Phonemic Level</u> - This level, named by seemingly contradicting terms, represents the utterance by phoneme-like units, with the addition of modifiers such as stress and boundary (word, morpheme, syllable) markings.

<u>Syllabic Level</u> - The unit of representation here is the syllable.

<u>Lexical Level</u> - The unit of information at this level is the word.

<u>Phrasal Level</u> - Syntactic elements appear at this level. In fact, since a level may contain arbitrarily many "sub-levels" of elements using the AND and OR links, traditional kinds of syntactic trees can be directly represented here.

<u>Conceptual Level</u> - The units at this level are "concepts." As with the phrasal level, it may be appropriate to use the graph structure of the data base to indicate relationships among different concepts.

As examples of knowledge sources, Figure 2 shows the first set implemented for HearsayII. The levels are indicated as horizontal lines in the figure and are labeled at the left. The knowledge sources are indicated by arcs connecting levels; the starting point(s) of

[1] Appendices A and B are reprinted from Lesser, et al (1974); they are included here for convenience.

an arc indicates the level(s) of major "input" for the KS, and the end point indicates the "output" level where the knowledge source's major actions occur. In general, the action of most of these particular knowledge sources is to create links between hypotheses on its input level(s) and: 1) existing hypotheses on its output level, if appropriate ones are already there, or 2) hypotheses that it creates on its output level.

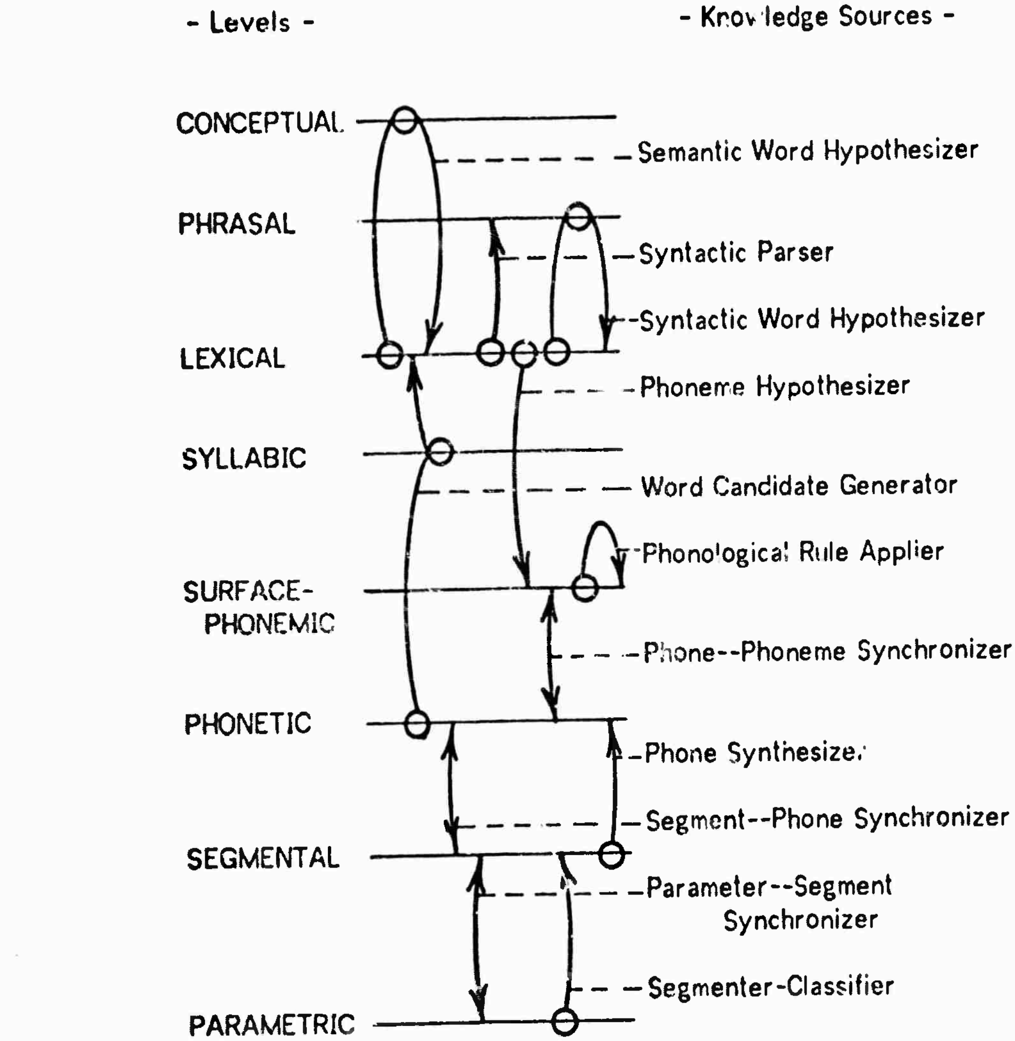


Figure 2. A Set of Knowledge Sources for HearsayII.

The Segmenter-Classifier knowledge source uses the description of the speech signal to produce a labeled acoustic segmentation. For any portion of the utterance, several possible alternative segmentations and labels may be produced.

The Phone Synthesizer uses labeled acoustic segments to generate elements at the phonetic level. This procedure is sometimes a fairly direct renaming of an hypothesis at the segmental level, perhaps using the context of adjacent segments. In other cases, phone synthesis requires the combining of several segments (e.g., the generation of [t] from a segment of silence followed by a segment of aspiration) or the insertion of phones not indicated directly by the segmentation (e.g., hypothesizing the existence of an [l] if a vowel seems velarized and there is no [l] in the neighborhood). This KS is triggered whenever a new hypothesis is created at the segmental level.

The Word Candidate Generator uses phonetic information (primarily just at stressed locations and other areas of high phonetic reliability) to generate word hypotheses. This is accomplished in a two-stage process, with a stop at the syllabic level, from which lexical retrieval is more effective.

The Semantic Word Hypothesizer uses semantic and pragmatic information about the task (news retrieval, in this case) to predict words at the lexical level.

The Syntactic Word Hypothesizer uses knowledge at the phrasal level to predict possible new words at the lexical level which are adjacent (left or right) to words previously generated at the lexical level. This knowledge source is activated at the beginning of an utterance recognition attempt and, subsequently, whenever a new word is created at the lexical level.

The Phoneme Hypothesizer knowledge source is activated whenever a word hypothesis is created (at the lexical level) which is not yet supported by hypotheses at the surface-phonemic level. Its action is to create one or more sequences at the surface-phonemic level which represent alternative pronounciations of the word. (These pronounciations are currently pre-specified as entries in a dictionary.)

The Phonological Rule Applier rewrites sequences at the surface-phonemic level. This KS is used: 1) to augment the dictionary lookup of the Phoneme Hypothesizer, and 2) to handle word boundary conditions that can be predicted by rule.

The Phone-Phoneme Synchronizer is triggered whenever an hypothesis is created at either the phonetic or the surface-phonemic level. This KS attempts to link up the new hypothesis with hypotheses at the other level. This linking may be many-to-one in either direction.

The Syntactic Parser uses a syntactic definition of the input language to determine if a complete sentence may be assembled from words at the lexical level.

The primary duties of the Segment-Phone Synchronizer and the Parameter-Segment Synchronizer are similar: to recover from mistakes made by the (bottom-up) actions of the Phone Synthesizer and Segmenter-Classifier, respectively, by allowing feedback from the higher to the lower level.

In addition to the knowledge source modules described above, all of which embody speech knowledge, several policy modules exist. These modules, which interface to the system in a manner identical to the speech modules, execute policy decisions, e.g., propagation of ratings and calculation of processing-state attributes.

Appendix B:
EXAMPLE OF A BLACKBOARD FRAGMENT IN HEARSAY II

Figure 3 is an example of a fragment that might occur in HearsayII's blackboard. The level of an hypothesis is indicated by its vertical position; the names of the levels are given on the left. Time location is approximately indicated by horizontal placement, but duration is only very roughly indicated (e.g., the boxes surrounding the two hypotheses at the phrasal level should be much wider). Alternatives are indicated by proximity; for example, 'will' and 'would' are word hypotheses covering the same time span. Likewise, 'question' and 'modal-question', 'you1' and 'you2', and 'J' and 'Y' all represent pairs of alternatives.

This example illustrates several features of the data structure:

The hypothesis 'you', at the lexical level, has two alternative phonemic "spellings" indicated; the hypotheses labeled 'you1' and 'you2' are nodes created, also at the lexical level, to hold those alternatives. In general, such sub-levels may be created arbitrarily.

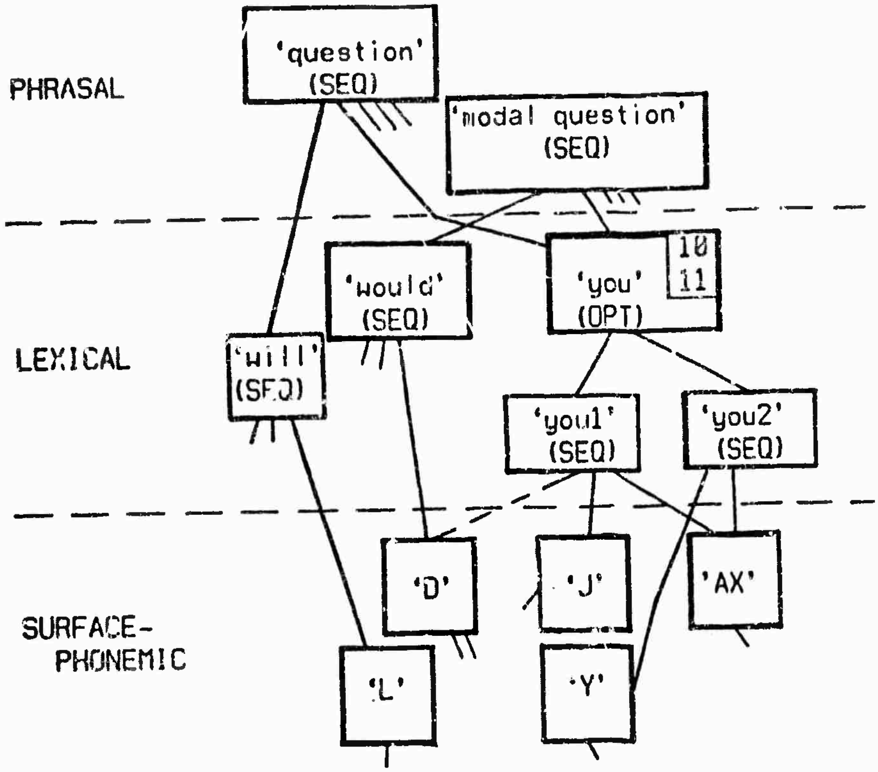


Figure 3. An Example of a Fragment in the Blackboard.

The link between 'you1' and 'D' is a special kind of SEQUENCE link (indicated here by a dashed line) called a <u>CONTEXT</u> link; a CONTEXT link indicates that the lower hypothesis supports the upper one and is contiguous to its brother links, but it is not "part of" the upper hypothesis in the sense that it is not within the time interval of the upper hypothesis -- rather, it supplies a context for its brother(s). In this case, one may "read" the structure as stating "'you1' is composed of 'J' followed by 'AX' (schwa) in the context of the preceding 'D'." (This reflects the phonological rule that "would you" is often spoken as "would-ja.") Thus, a CONTEXT link allows important contextual relationships to be represented without violating the implicit time assumptions about SEQUENCE nodes.

Whereas the phonemic spelling of the word "you" held by 'you1' includes a contextual constraint, the 'you2' option does not have this constraint. However, 'you1' and 'you2' are such similar hypotheses that there is strong reason for wanting to retain them as alternative options under 'you' (as indicated in Figure 3), rather than representing them unconnectedly. A connection matrix is used here to represent this kind of relationship; the connection matrix of 'you' (symbolized in Figure 3 by the 2-dimensional binary matrix in the node) specifies that support 'you1' is relevant to use 'question' (but not to 'modal-question') and that support 'you2' is relevant to both uses.

The nature of the implications represented by the links provides a uniform basis for propagating changes made in one part of the data structure to other relevant parts without necessarily requiring the intervention of particular knowledge sources at each step. Considering the example of Figure 3, assume that the validity of the hypothesis labeled 'J' is modified by some KS (presumably operating at the phonetic level) and becomes very low. One possible scenario for rippling this change through the data base is given here:

First, the estimated validity of 'you1' is reduced, because 'J' is a lower hypothesis of 'you1.'

This, in turn, may cause the rating of 'you' to be reduced.

The connection matrix at 'you' specifies that 'you1' is not relevant to 'modal-question,' so the latter hypothesis is not affected by the change in rating of the former. Notice that the existence of the connection matrix allows this decision to be made locally in the data structure, without having to search back down to the 'D' and 'J.'

'Question,' however, is supported by 'you1' (through the connection matrix at 'you'), so its rating is affected

Further propagations can continue to occur, perhaps down the other SEQUENCE links under 'question' and 'you1.'

Notice that all of these modifications are "speech-knowledge independent" and can be accomplished uniformly at all levels of the blackboard by a single policy knowledge source. This policy KS does not need to access or trigger any other KS but can directly derive all the information it needs from the hypothesis and link fields that are uniformly present and from the implicit semantics of the structures in the blackboard.

REFERENCES

Baker, James (1974), "The DRAGON System -- An Overview," in Erman (1974b).

Barnett, J. (1973), "A Vocal Data Management System," **IEEE Trans. Audio and Electroacoustics,** AU-21, 3.

Barnett, J. (1975, in press), "Module Linkage and Communication in Large Systems," in Reddy (1975).

Erman, L. D., R. D. Fennell, V. R. Lesser, and D. R. Reddy (1973), "System Organizations for Speech Understanding: Implications of Network and Multiprocessor Computer Architectures for AI," **Proc. 3rd Inter. Joint Conf. on Artificial Intel.,** Stanford, Ca., pp. 194-199.

Erman, L. D. (1974), "An Environment and System for Machine Understanding of Connected Speech," Ph.D. thesis, Comp. Sci. Dept, Stanford Univ.

Erman, L. D. (ed.) (1974b), **Contributed Papers of the IEEE Symposium on Speech Recognition,** April 15-19, 1974, Pittsburgh, Pa., IEEE Cat. No. 74CH0878-9AE. Many of these papers have been reprinted in **IEEE Trans. on Acoustics, Speech, and Signal Processing,** ASSP-23, no. 1 (Feb., 1975).

Fennell, R. D., and V. R. Lesser (1975), "Parallelism in A.I. Problem-Solving: A Case Study of HearsayII," Tech. Report, Comp. Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pa.

Fennell, R. D. (1975b, in preparation), Ph.D. thesis, Comp. Sci. Dept., Carnegie-Mellon Univ.

Hayes-Roth, F., V. R. Lesser, and D. W. Kosy (1975, in preparation), "A Design for Attentional Control in a Distributed-Logic Speech Understanding System," Tech. Report, Comp. Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pa.

Hewitt, C. (1972), "Description and Theoretical Analysis (Using Schemata) of Planner: A Language for Proving Theorems and Manipulating Models in a Robot," AI Memo No. 251, MIT Project MAC.

Lesser, V. R., R. D. Fennell, L. D. Erman, & D. R. Reddy (1974), "Organization of the HEARSAY II Speech Understanding System," in Erman (1974b). Also appeared in **IEEE Trans. on Acoustics, Speech, and Signal Processing,** ASSP-23, no. 1, pp. 11-23 (Feb., 1975).

Lesser, V. R. (1975, in press), "Parallel Processing in Speech Understanding Systems: A Survey of Design Problems," in Reddy (1975).

Nowoll, A. (1969), "Heuristic Programming: Ill-Structered Prob. ms," in J. S. Aronofsky (ed.), **Progress in Operations Research, 3**, Wiley, 363-415.

Newell, A., J. Barnett, J. Forgie, C. Green, D. Klatt, J. C. R. Licklider, J. Munson, R. Reddy, and W. Woods (1971), **Speech Understanding Systems: Final Report of a Study Group**, pub. by North-Holland (1973).

Newell, A. (1973), "Production Systems: Models of Control Structures," in W. C. Chase (ed.) **Visual Information Processing**, Academic Press, pp. 463-526.

Reddy, D. R. (1966), "An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave," (Ph.D. thesis) AI Memo No. 43, Comp. Sci. Dept., Stanford Univ., Stanford, Ca.

Reddy, D. R., L. D. Erman, and R. B. Neely (1973a), "A Model and a System for Machine Recognition of Speech," **IEEE Trans. Audio and Electroacoustics**, AU-21, 3, pp. 229-238.

Reddy, D. R., L. D. Erman, R. D. Fennell, and R. B. Neely (1973b), "The HEARSAY Speech Understanding System: An Example of the Recognition Process," **Proc. 3rd Inter. Joint Conf. on Artificial Intel.**, Stanford, Ca., pp. 185-193.

Reddy, D. R. (1973), "Eyes and Ears for Computers," Tech. Report, Comp. Sci. Dept., Carnegie-Mellon University, Pittsburgh, Pa. Keynote speech presented at Conf. on Cognitive Processes and Artifical Intelligence, Hamburg, April, 1973.

Reddy, R., and A. Newell (1974), "Knowledge and its Representation in a Speech Understanding System," in L. W. Gregg (ed.) **Knowledge and Cognition**, Lawrence Erlbaum Assoc., Washington, D. C., chap. 10, (in press).

Reddy, D. R. (ed.) (1975, in press), Invited **Papers of the IEEE Symposium on Speech Recognition**, April 15-19, 1974, Pittsburgh, Pa., Academic Press.

Rovner, P., Nash-Webber, B., and Woods, W. (1974), "Control Concepts in a Speech Understanding System," in Erman (1974b). Also appeared in **IEEE Trans. on Acoustics, Speech, and Signal Processing**, ASSP-23, no. ., pp. 136-140 (Feb., 1975).

Rulifson, J. F., et al., (1973), "QA4: A Procedural Calculus for Intuitive Reasoning," Technical Note 73, AI Center, Stanford Res. Inst., Menlo Park, Ca.

Simon, H. A., (1962), "The Architecture of Complexity", **Proc. Amer. Philosophical Society**, 105, pp. 467-482.

Vicens, P. (1969), "Aspects of Speech Recognition," Report CS-127, (Ph.D. Thesis), Comp. Sci. Dept., Stanford Univ.