A PARALLEL PARSER FOR SPOKEN NATURAL LANGUAGE *

Egidio P. GIACHIN and Claudio RULLENT

CSELT - Centro Studi e Laboratori Telecomunicazioni S.p.A
Via Reiss Romoli, 274 - 10148 Torino - Italy

ABSTRACT

This paper describes SYNAPSIS, a parser for performing real-time understanding of spoken utterances in a parallel computational environment. Understanding continuous speech allowing reasonably free syntax poses two main oroblems, namely the risk of erroneous interpretations and the largeness of the search space owing to the high uncertainty of the input. The parser is characterized by an approach whose major novel features are 1) a drastic reduction of idle time, thanks to the asynchronous inter agent communications, and 2) high modularity, thanks to the distribution of homogeneous pieces of knowledge rather than distribution of different parsing tasks. These features make the parser most apt for implementation on homogeneous Transputer-based distributed architectures.

1. Introduction

This paper describes SYNAPSIS, the parallel parser that is being developed at CSELT Laboratories. SYNAPSIS (from SYNtax-Aided Parser for Semantic Interpretation of Speech) is part of a question-answering system [Fissore et al., 1988] that allows to extract information from a data base using voice messages with high syntactic freedom. The whole system is composed of a recognition stage which analyzes continuous speech utterances and produces a set of word hypotheses (usually called lattice in the literature); and of an understanding stage which analyzes the word lattice using linguistic knowledge and produces a representation of the meaning of the most likely consistent word sequence.

The major aspect that distinguishes spoken from written language parsing lies in the type of input. The current state-of-art in large vocabulary continuous speech processing does not permit the unique identification of the uttered words if no language constraints are used along with constraints derived from acoustical and phonetical knowledge. In cases like the one we are addressing, where language is expressed by a large and complex grammar, it would be impractical to use language constraints at the recognition level. Thus what is actually provided to the understanding stage is not a sequence of words but a set of word hypotheses (WHs). A Word Hypothesis is a structure with three elements: an acoustical quality factor, or score for brevity, representing its belief degree; a temporal interval representing the location in the utterance where it was spotted; and the lexeme itself. The Word Hypotheses represent highly uncertain data; 'spurious' Word Hypotheses heavily outnumber correct Word Hypotheses and some are even better scored. Moreover, there often exist different subsets of Word Hypotheses that correspond to plausible, but not really uttered, sentences.

Summarizing, the additional problems posed by speech vs. text parsing are the following:

1) High risk of erroneous understanding due to the presence of spurious Word Hypotheses;
2) Very large search space, because the non-determinism typical of natural language parsing is added to the non-determinism arising from the uncertainty of the input.

One first requirement to cope with the above problems is to seek a parsing strategy that is intrinsically efficient and easily driven by the Word Hypothesis scores. However, this is not enough to reach real-time performances, especially when the system is intended to gradually increase the size of its lexicon and/or linguistic coverage. In such case the parser must do its activity on a highly modular, easily extendable parallel processing environment.

SYNAPSIS derives from the sequential parser employed in the SUSY system [Poesio and Rullent, 1987], which was designed with a special emphasis on efficient integration of high-level syntactic/semantic knowledge within an opportunistic analysis strategy. SYNAPSIS departs from other parallel parsers in one fundamental respect: what is partitioned among the various agents is the total linguistic knowledge and not the different parsing tasks. That is. every agent has the whole inferencing capability of the sequential parser; only, it relies on a fraction of the total knowledge.The type of parallelism resulting from such a multi-agent organization is a kind of OR-parallelism on the active nodes of partial parses. The advantages of this scheme are:

1) Asynchronicity. The communications among agents are asynchronous, that is there are no 'wait-for-reply' idle times. Each agent has its own agenda of pending tasks that refers to the generation of phrase hypotheses pertaining to the linguistic knowledge of the agent itself.

* This work has been partially supported by the EEC within Esprit projects P26 and P967

2) Ease of implementability. The data-flow nature of the algorithm is directly mirrored by homogeneous Transputer-based distributed architectures supporting languages endowed with asynchronous message-passing primitives. A description of what a global (hardware + algorithmic) architecture should result is given in [Bosco et al. 1987].

3) Modularity. To increase the knowledge base it is sufficient to introduce new agents or to distribute the new pieces of knowledge among existing agents.

The sequential parser has been experimentally proven to lie in the state-of-the-art in speech processing [Giachin and Rullent, 1988, DeMattia and Giachin, 1989]. The feasibility and effectiveness of the distributed parser has been demonstrated not only by simulation but also by implementing it on a pool of workstations working in parallel; alongside, the parallel hardware is being developed. In the following the principles of sequential parsing are briefly resumed, prior to the description of the parallel parser. Further discussion on the position of the parallel parser with respect to the current research scenario is given in Section 4.

2. The sequential parsing strategy

2.1 Summary of the recognition stage

In the recognition stage words are defined as sequences of elementary speech units. Such units are represented by Hidden Markov Models (HMMs) of speech spectral properties. The Forward decoding algorithm is used to hypothesize words along the utterance and to assign them a score that expresses how close is the match between the ideal word model and the actual observed utterance portion. The HMM technique is widely used for large vocabulary speech recognition and gives comparable results for different languages [Fissore et al.. 1989, Lee 1988].

2.2 Linguistic knowledge representation

The linguistic knowledge representation used by the system is based on the notion of caseframe [Fillmore, 1968]. Caseframes allow to describe a language's semantics in a compositional way, through a header word (usually a verb or a noun) and a set of cases that may in turn correspond to other caseframes. Caseframes offer several advantages in speech parsing, including a way of correlating semantic significance with acoustic certainty. This happens because the header word, being the most "meaningful" one, tends to be uttered more clearly, and hence is easily recognized with good acoustical score. This explains their popularity in many recent speech understanding systems [Hayes et al., 1986, Brietzmann and Ehrlich, 1986, Niedermair, 1986].

Caseframe-based speech parsing raises two difficulties, however. One is that parsing is induced to proceed by instantiating caseframes in a top-down fashion. This causes severe problems if the uttered sentence includes a bad-scored word that constitutes the header of a caseframe, because that frame will not be resumed until all better-scored items have been examined, thus spurring a lot of useless search activity. The second difficulty is the integration of caseframes with syntax. For many reasons (ease of development and maintenance, possibility of using flexible representational formalisms) syntax should be defined and developed separately from caseframes, but to reduce the size of the search activity syntactic constraints should be used together with semantic constraints during parsing. This idea, though under differerent perspectives, permeates much of current speech understanding research [Tomabechi and Tomita, 1988, Hauptmann et al. 1988].

To overcome these problems, caseframes and syntax are pre-compiled into structures called Knowledge Sources (KSs). Each Knowledge Source owns the syntactic and semantic competence necessary to get through a well-formed interpretation of a fragment of the input. Fig. 1 shows a caseframe (represented via Conceptual Graphs [Sowa, 1984]) and the resulting Knowledge Source obtained by combining it with two rules of a Dependency Grammar [Hays, 1964]. Note that the 'compositional' part of the Knowledge Source directly mirrors the phrase structure expressed by syntactic rules in the form of immediate constituents. The Knowledge Sources do not form a classical context-free semantic grammar, however. Beside the 'compositional' part, in fact, the Knowledge Sources possess additional information which is used to establish constraints of both semantical and syntactical nature on the different constituents. These constraints account, for instance, for morphological agreement between distant words (of which some languages, like Italian, are particularly rich). This property, together with computationally fast methods for propagating and checking such constraints, permits to enormously reduce the number of Knowledge Sources that would treat very similar language constructs. Analogous principles are currently tested in systems based on unification grammars [Chow and Roucos, 1989].

```
[LOCATED-IN-REGION]
   -->  (AGNT:Compulsory) --> [MOUNT+PROVINCE+LAKE]
   -->  (LOC:Compulsory)    --> [REGION]

KS24.2:

  ;; Compositional part
 LOCATED-IN-REGION = REGION <header>
                          MOUNT+PROVINCE+LAKE

  ;; Associated Syntactic Rules
 (DR24 DR24A)
;;; VERB = NOUN <governor> NOUN
;;; VERB = NOUN <governor> PROPER-NOUN

  ;;; Activation Condition
 G(?x) ==> ACTION (?x  LOCATED)

  ;;; Caseframe Instantiation Rule and Meaning
  ;;; Representation
 *RIS((LOCATED  ! * AGNT  ?z  LOC  ?y)) <==
    RIS(REGION  (ACTION  !  *  ?x  ?z))
    RIS(MOUNT+PROVINCE+LAKE  (ACTION  !  *  ?w  ?y)))
```

Fig. 1 - A Caseframe in the Conceptual Graph notation and a corresponding Knowledge Source.

2.3 Parsing

Parsing consists in letting phrase hypotheses be constructed by Knowledge Sources until the whole utterance is covered. Phrase hypotheses are an extension of the 'island' concept introduced in past speech literature [Woods, 1982], the major difference being that they do not require their component Word Hypotheses to be contiguous, but only to make up a consistent (possibly incomplete) parse tree. To each phrase hypothesis an acoustic score is assigned, computed from the likelihood score of its Word Hypotheses. An example is shown in Fig. 2. It required three Knowledge Sources and still has a 'goal' node in it (i.e. a node for which no Word Hypotheses were found yet).

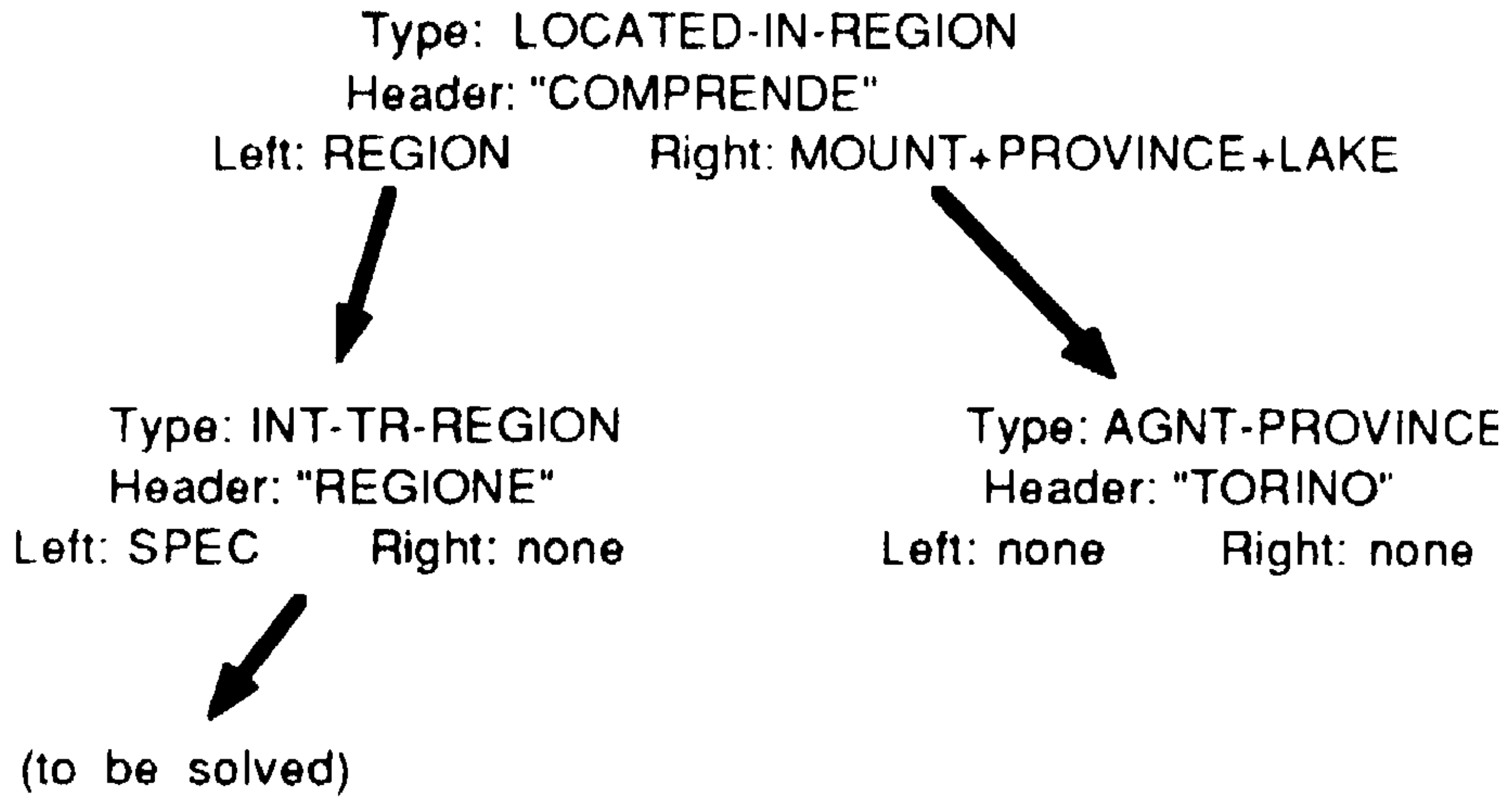

Fig. 2 - An example of perse tree, accounting for the sentence "Quale regione comprende Torino?" ("Which region includes Turin?"). "Quale" has not yet been found. The root of this parse tree was build by the Knowledge Source of Fig.1.

Parsing follows a best-first strategy. At each cycle, the best-scored item is selected (either a phrase hypothesis or a word hypothesis) and it is accreted with other word or phrase hypotheses by all of the Knowledge Sources that can do the job. The result is the production of new, bigger phrase hypotheses.The elementary actions on phrase or word hypotheses are described by operators. Top-down, prediction-based actions are dynamically mixed with bottom-up, expectation-based actions.

Top-down actions consist in starting from a phrase hypothesis whose parse tree has an empty node, and:

1) fill it with word hypotheses (VERIFY operator) if it is a header node (which is a terminal one), or
2) fill it with already existing complete parse trees (MERGE) if it is a case node (a nonterminal one), or still
3) decompose it accordingly to the compositional structure of a Knowledge Source (SUBGOALING).

Bottom-up actions consist in creating a phrase hypothesis starting from a word hypothesis, which will occupy the header node of the newly-created phrase hypothesis (ACTIVATION), or starting from a complete parse-tree, which will occupy one of the case nodes (PREDICTION, MERGE). The actions to be performed on the selected item are determined solely by its characteristics, accordingly to the above strategy.

Such a parsing strategy permits to use both headers to predict words at lower levels and vice versa. Consequently, any well-scored phrase hypotheses is guaranteed to be treated and expanded with other words, whether this can be done with a bottom-up or with a top-down step. This eliminates the bottlenecks that occur with standard top-down caseframe parsing, while preserving most of the advantages that caseframes offer over semantic grammars in terms of flexibility.

3. The distributed parser

The parallel parser is obtained by distributing the Knowledge Sources among N agents called Distributed Problem Solvers (DPSs). Each DPS has the whole inferencing capability of the sequential parser, but relies only on a subset of Knowledge Sources to perform its activity. Since every single parse tree requires in general more than one Knowledge Source to be built, and the required Knowledge Sources may reside on different DPSs, it follows that the various DPSs must cooperate to carry on the overall analysis. This could be achieved by adopting a blackboard architecture. However, a blackboard architecture proves unsatisfactory when strict control on activity scheduling is required [Corkill et al. 1982], like the one outlined in the previous section. Hence an approach has been followed in which the partial parses are dismembered into smaller elements that are distributed among the DPSs. Each DPS has a private data base of active elements, each one representing a whole parse tree, on which the control strategy described above is applied. How this is accomplished is described in the next sections.

3.1 Distributing the parse trees

The basic concept consists in 'dismembering' each parse tree into one-level subtrees, called Physical Hypotheses (PHs) according to Fig. 3. One of the Physical Hypotheses - called the active one - is able to represent

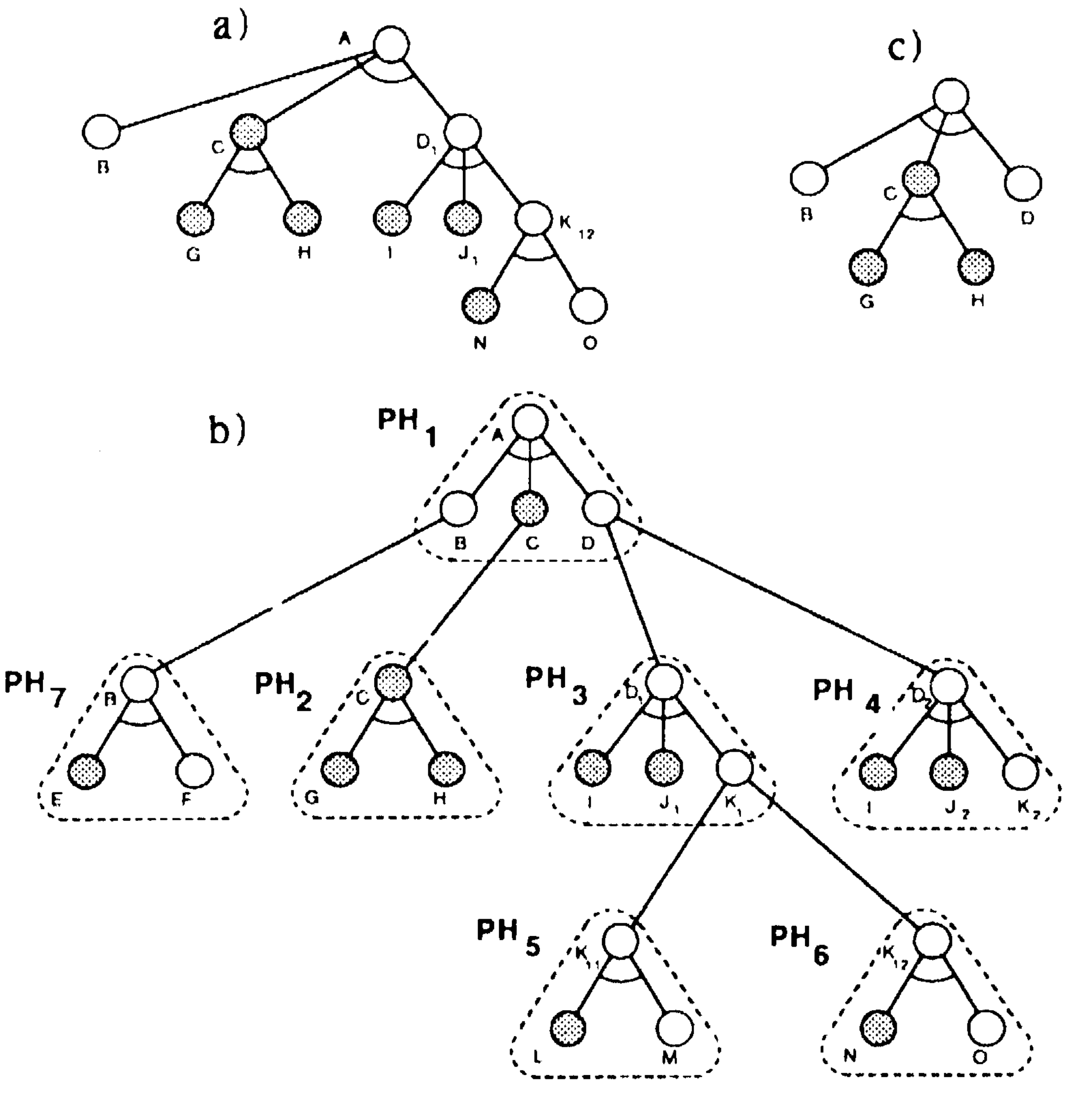

Fig. 3 - Representation of parse trees in the distributed system.
a) A parse tree (filled circles represent complete items).
b) The same parse tree separed into Physical Hypotheses. The active one is PH6. This structure represents other parse trees that share common subtrees.
c) A parse tree represented by the structure depicted in b) via PH1 as the active Physical Hypothesis.

the whole parse tree as far as an operator application is concerned. It carries a priority equal to the acoustical score of the parse tree it represents. By comparing Fig. 3 with Fig. 2 it is seen that, since a Physical Hypothesis is a one-level tree, it directly corresponds to the compositional part of some Knowledge Source. In other words, each Physical Hypothesis expresses a hypothesis on the phrase structure composition of a segment of utterance. It follows that the Physical Hypotheses can be partitioned according to their corresponding Knowledge Source and consequently to the DPS on which the Knowledge Source resides.

3.2 Parsing and control in the parallel system

The architecture of a DPS is depicted in Fig. 4. The DPSs cooperate by exchanging relevant information about the active Physical Hypotheses corresponding to the parse trees they are treating. For example, if a DPS has generated a complete parse tree that could be used by another DPS to predict other parses (this corresponds to an application of the PREDICTION operator), a set of information about the Physical Hypothesis representing that complete tree is sent to the interested DPS; the recipient DPSs will generate pending tasks whose priority depends on the score of that complete parse tree.

At the beginning of the analysis of an utterance, the lattice of Word Hypotheses generated by the recognition system is first distributed among the DPSs. The criterion for distributing the lattice simply consists in assigning to a DPS the Word Hypotheses whose lexeme can be used as a header by at least one of the local Knowledge Sources. This requires some duplication of Word Hypotheses in different DPSs, depending on the way Knowledge Sources have been partitioned.

At this point the actual analysis begins. The exchanged information is embodied into messages representing requests for some actions. The delivering DPS does not enter a 'wait' state in expectance of a reply from the recipient DPS: that is, communication is asynchronous at all times. The messages, once received, are scheduled according to the priority of the parse tree they refer, so that the recipient DPS only executes the highest-priority messages first. Similarly, the delivering DPS can continue to work after sending its message by treating the other Physical Hypotheses with the highest priorities; in other words, the parsing path that led to generating the message comes to a temporary stop, and old parsing paths are resumed or new ones are started. This drastically reduces (in theory eliminates) idle times.

Table I illustrates the different messages that a single DPS deals with, together with the operators they activate and other relevant information. The first three messages (Subgoal Resolution Request, Prediction-Merging Request, and Subgoal Resolution Answer) are in general remote messages, that is messages coming from other DPSs. The first two messages roughly correspond to respectively top-down and bottom-up activities. The last one simply informs a DPS that a subgoal it had previously asked to solve has indeed been solved; the involved activities consist only in creating a Physical Hypothesis representing the new parse tree. This message is not subjected to priority scheduling and is executed as soon as possible when the current scheduling cycle is terminated. The other two messages (Verify and Activation) are local.

The first two messages, Subgoal Resolution Request and Prediction-Merging Request, are subjected to the local control scheduler, together with Verify and Activation (see Fig. 4). The interesting aspect is that, since every message refers directly to a particular Physical Hypothesis and hence a particular parse tree, it can be assigned a

Received Request	Performed Activities	Activated Operators	New parse trees	Request that will be delivered
SRR *(Remote)*	1) Merging incomplete parses with complete ones	MERGE	Complete, Incomplete	SRA
Subgoal Resolution Request	2) Decomposing a non terminal node	SUBGOALING	Incomplete	VR
PMR *(Remote)* Prediction-Merging	1) Prediction of incomplete parses from complete ones	PREDICTION	Incomplete	VR
Request	2) Merging a complete parse with an incomplete one	MERGE	Incomplete Complete	SRR or VR PMR or SRA
SRA *(Remote)* Subgoal Resolution Answer	Acquisition of the complete parse	---	Incomplete Complete	SRR or VR PMR or SRA
VR *(Local)* Verify Request	Solution of a header node	VERIFY	Incomplete Complete	SRR PMR or SRA
ACT *(Local)* Activation	Activation of a Word Hypothesis	ACTIVATION	Incomplete Complete	SRR PMR or SRA

Table 1 - The messages exchanged between DPSs and the associated activities

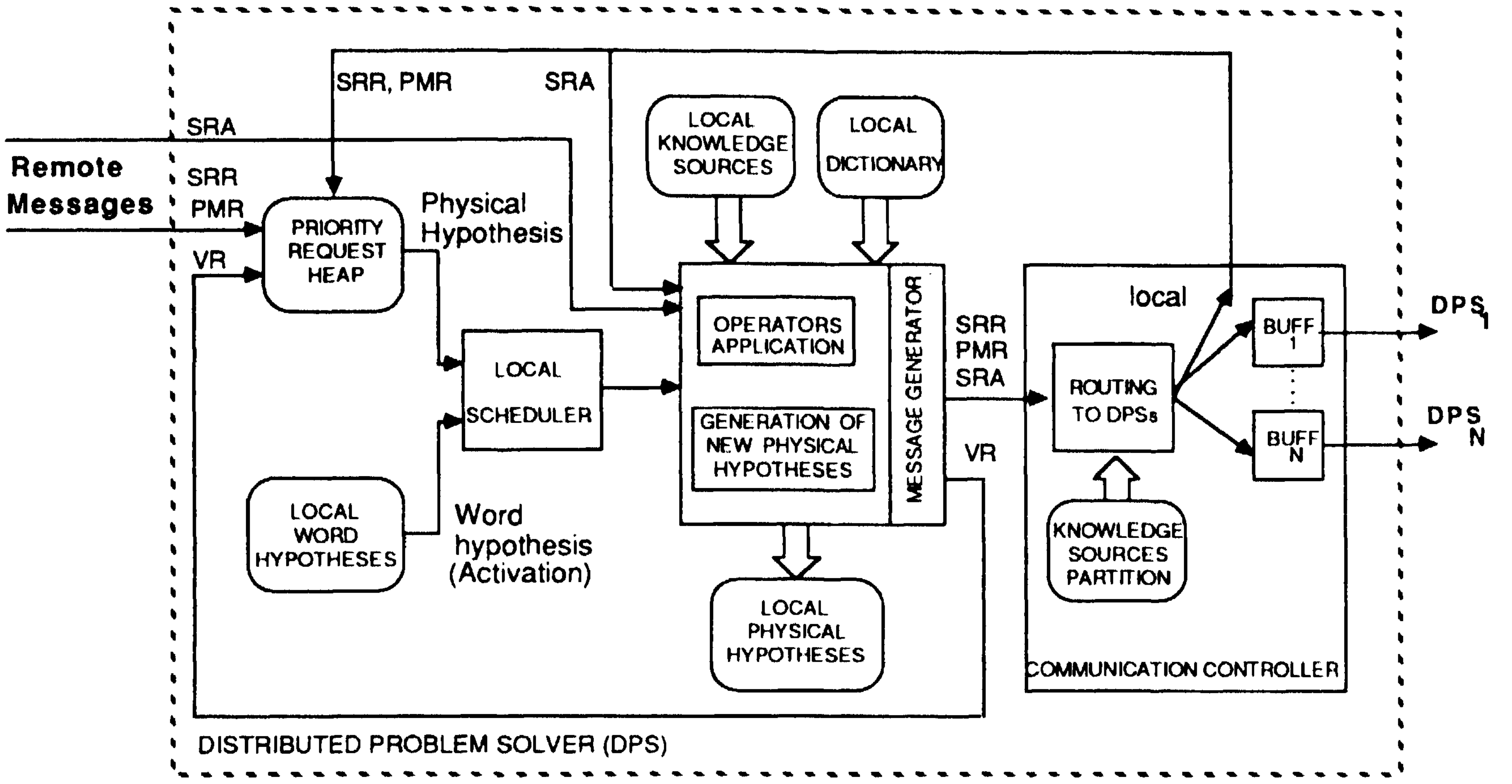

Fig. 4 - Architecture of a Distributed Problem Solver.

priority equal to the score of the parse tree it corresponds to; then the scheduling allows the kind of score-guided control previously outlined. Specifically, the message is selected whose referent parse tree is characterized by the highest score. The Physical Hypothesis corresponding to that parse tree can be local to the DPS or not. In this way, at any time, the best-scored (in absolute) parse tree is always guaranteed to be treated by one of the DPSs.

The local scheduler is also faced with the Activation phase. When there is a Word Hypothesis (among those that were initially assigned to the local DPS) with a better score than the best priority of the messages contained in the Priority Request Heap, that Word Hypothesis is used to perform the Activation phase. In this phase the Word Hypothesis is used to solve the headers (which are terminal nodes) of all the Physical Hypotheses residing on the DPS. This activity directly corresponds to the ACTIVATION operator.

Whenever a new Physical Hypothesis is created, a corresponding request is generated, according to Table I. This request has the purpose of making this new Physical Hypothesis visible by other DPSs that could treat it. A Communication Controller generates the messages to be sent to the involved DPSs. That assumes that there must be knowledge about what DPS can treat the newly generate Physical Hypothesis. This knowledge comes from the given partition of the Knowledge Sources among the DPSs and is stored in the Communication Controller.

4. Implementation and related research

The effectiveness of the speech-parsing approach described throughout Section 2 has been demonstrated through several experimentations on the SYNAPSIS version running sequentially. The whole speech understanding system correctly recognizes and understands 80% of continuously-uttered sentences, on the average 7 words long, with a dictionary of 1000 words and a language knowledge represented by about 200 Knowledge Sources (that corresponds to a language model with branching factor about 35). The analysis of one sentence takes on the average 40 seconds on a SYMBOLICS.

The current working parallel implementation of SYNAPSIS runs on a pool of Lisp Machines, each acting as a DPS. This solution, hardly the cheapest one, offers enormous advantages in terms of ease of development and testing, though does not permit specific parallel-algorithm measures (for instance, measuring the time spent in communication is not significant since the processors can only communicate via Ethernet). A future hardware environment will consist of a Transputer-based distributed architecture.

In the present view, each DPS is statically allocated on a processor. A possible drawback of this static, processes as Knowledge Sources', model is that it may result in a poor utilization factor of the available processing resources, in contrast with a dynamic, processes as hypotheses', model, that has been rejected (at least for the time being) due to the high communication rate required and the great implementative complexities. On the other hand, we have experimentally verified that, using a suitable partition of the Knowledge Sources, each processor produces about the same number of hypotheses, thus signifying that a satisfactory exploitation of parallelism can be achieved in practice. Moreover, limited processing resource utilization within a single system could be not a critical problem if a system-level parallelism (due to multiplexing input data) is guaranteed to exist in a final application, as could happen, for example, in multi-user telecommunication equipments for voice-based advanced services.

The amount of communication among the DPSs depends on the way the Knowledge Sources are partitioned among the DPSs. A suitable partition can keep it small as compared with the activities performed at every request on the basis of local information. In order to furtherly reduce it, the communication output buffers can be managed taking into account the values of the best priorities of the DPSs to which the messages are sent. This requires that the current priority be periodically communicated by each DPS to all of the others, so that the communication rate is decreased from one side but increased from the other. In the present implementation this feature has not been included.

The recent literature on parallel parsing is relatively limited. Moreover, it seems that written language has been privileged over spoken one. This explains why most works address different issues from our own. These include efficient distributing of parsing tasks within schemes able to deal with a range of high-level linguistic phenomena [Huang and Guthrie, 1986. Eiselt, 1985], or investigation of parallel approaches under paradigms related to the connectionist model [Pollack and Waltz. 1985, Slack 1986], or still master-directed parallelization and elaboration of algorithms for existing formalisms [Haas, 1987]. In contrast, the parser described here is intended to face problems deriving from the high ambiguity at the input level, which is the salient characteristic of speech. To allow effective 'multiple attack' to the input, emphasis is given to distributing Knowledge Sources rather than tasks, and to adopting a cooperating-agent framework able to perform a score-guided analysis without relying on synchronous, deadlock-prone form of communication.

References

[Bosco et al., 1987] P.Bosco, E.Giachin, G.Giandonato, G.Martinengo, C.Rullent. A Parallel Architecture for Signal Understanding through Inference on Uncertain Data. *Lecture Notes in Computer Science,* Vol. 258, pages 86-102. Springer-Verlag, 1987.

[Brietzmann and Ehrlich, 1986) A.Brietzmann, U.Ehrlich. The role of semantic processing in an automatic speech understanding system. In Proc. *COLING-86,* Bonn, pp. 596-598.

[Chow and Roucos, 1989] Y.L.Chow, S.Roucos. Speech understanding using a unification grammar. In *Proc. ICASSP-89,* Glasgow.

[Corkill et al., 1982] D.D.Corkill, V.R.Lesser, E.Hudlicka. Unifying data-directed and goal-directed control: an example and experiments. In *Proc. AAAI-82,* Pittsburgh, pp. 143-147.

[DeMattia and Giachin, 1989] M.DeMattia, E.Giachin. Experimental results on syntactic-semantic processing for large vocabulary continuous speech recognition and understanding. In *Proc. ICASSP-89,* Glasgow.

[Eiselt, 1985] K.P.Eiselt. A parallel-process model for on-line inference processing. In *Proc. IJCAI-85,* Los Angeles, pp. 863-869.

[Fillmore, 1968] C.J.Fillmore. The case for case. In Bach, Harris (eds.), *Universals in Linguistic Theory,* Holt, Rinehart, and Winston, New York, 1968.

[Fissore et al., 1988] L.Fissore, E.Giachin, P.Laface, G.Micca, R.Pieraccini, C.Rullent. Experimental results on large-vocabulary speech recognition and understanding. In *Proc. ICASSP-88,* New York.

[Fissore et al., 1989]: L.Fissore, P.Laface, G. Micca and R.Pieraccini. A Word Hypothesizer for a Large Vocabulary Continuous SUS. In *Proc. ICASSP-89,* Glasgow.

[Giachin and Rullent, 1988] E.Giachin, C.Rullent. Robust parsing of severely corrupted spoken utterances. In *Proc. COLING-88,* Budapest, pp. 196-201.

[Haas, 1987] A.Haas. Parallel parsing for unification grammars. In *Proc. IJCAI-87,* Milano, pp. 615-618.

[Hauptmann et al., 1988] A.G.Hauptmann, S.R.Young, W.H.Ward. Using dialog-level knowledge sources to improve speech recognition. In Proc. AAAI-88, Saint Paul (Minnesota), pp. 729-733.

[Hayes et al., 1986] P.J.Hayes, A.G.Hauptmann, J.G.Carbonell, M.Tomita. Parsing spoken language: a semantic caseframe approach. In Proc. *COLING-86,* Bonn, pp. 587-592.

[Hays, 1964] D.G.Hays. Dependency theory: a formalism and some observations. Memorandum RM4087 P.R., The Rand Corporation.

[Huang and Guthrie. 1986J X.Huang, L.Guthrie. Parsing in parallel. In *Proc. COLING-86,* Bonn, pp. 140-145.

[Lee, 1988]: K.F.Lee. Large-vocabulary speaker-independent continuous speech recognition: the SPHINX system. Ph.D. Thesis, Comp. Sci. Department, Carnegie Mellon University, Pittsburgh, April 88.

[Niedermair, 1986] G.T.Niedermair. Divided and valency-oriented parsing in speech understanding. In *Proc. COLING-86,* Bonn. pp. 593-595.

[Pollack and Waltz, 1985] J.Pollack, D.Waltz. Massively parallel parsing: a strongly interactive model of natural language interpretations. *Cognitive Science 9.*

[Poesio and Rullent. 1987] M.Poesio. C.Rullent. Modified caseframe parsing for speech understanding systems. In Proc. *IJCAI-87,* Milano, pp. 622-625.

[Slack, 1986] J.M.Slack. Distributed memory: a basis for chart parsing. In *Proc. COLING-86,* Bonn, pp. 476-481.

[Sowa, 1984] J.F.Sowa. *Conceptual Structures.* Addison Wesley, Reading (MA), 1984.

[Tomabechi and Tomita, 1988] H.Tomabechi, M.Tomita. The integration of unification-based syntax/semantics and memory-based pragmatics for real time understanding of noisy continuous speech input. In *Proc. AAAI-88,* Saint Paul (Minnesota), pp. 724-728.

[Woods, 1982] W.A.Woods. Optimal search strategies for speech understanding control. *Artificial Intelligence,* 18: 295-326, 1982.