KNOWLEDGE BASED SYSTEMS FOR SPEECH UNDERSTANDING

G. Sagerer, F. Kummert
Lehrstuhl für Informatik 5 (Mustererkennung)
Universität Erlangen-Nürnberg
Martensstrasse 3
8520 Erlangen, F.R. Germany

Abstract: Based on a detailled discussion of system architectures for knowledge based speech understanding and knowledge representation techniques, criteria for both a knowledge representation scheme and system architectures are developed. Upon this background a system is introduced which is organized around a homogenuous knowledge base. Both the knowledge representation language and the content of the knowledge base are described. The knowledge representation language do not only cover declarative but also procedural knowledge. Analysis processes are guided by a flexibel bottom-up top-down strategy. Besides the procedural semantics of the language the A -Algorithm is used for this purpose. Search graph nodes are judged by a vector which reflects knowledge dependent and acoustic scores and which is admissable for the A -Algorithm.

1. INTRODUCTION

The term "knowledge based speech understanding" is strongly related to the first ARPA Speech Understanding Reseach Project /LEA 80a/. A number of different system architectures were developed and applied to speech understanding tasks. Architectures like the blackboard model of Hearsay-II /ERM 80a/ were also adopted to other problems like Computer Vision /KIM 84a/. But independent of speech applications, knowledge based systems were studied with great effort in the last years. Especially the technology of Expert Systems /HAY 82a, WAT 86a/ influenced the development of systems in the speech recognition /DEM 83a, MER 85a/ and understanding area /MCC 81a, NIE 86a/. The progress in knowledge representation resulted in different powerful approaches /MCC 83a/. Schemes and languages developed in this area are of increasing influence on natural language and speech understanding systems /THU 88a/. Besides these successful mutual effects between speech understanding systems and knowledge based

NATO ASI Series, Vol. F46
Recent Advances in Speech Understanding and Dialog Systems
Edited by H. Niemann et al.

systems in general, a gap arose during the last years inside the speech understanding community. Perhaps based on the success of the Harpy /LOW 80a/ and the IBM /BAH 83a/ systems concerning the recognition rate of spoken texts, the recognition and the understanding of speech were more and more treated as two sequential, not cooperating and not interacting tasks /RAB 88a, SCH 88a/.

All these systems are based on a left to right analysis of the input signal. Linguistic knowledge is precompiled and implicitly used during the recognition process. If we change from spoken texts to dialog oriented systems, analysis becomes more complicated. Background of the following investigations to this problem is the system EVAR. EVAR is the abbreviation of the German words for recognition, understanding, answering, asking back. It is designed as speech understanding and dialog system with IC-train schedule information as discourse domain. Additionally, EVAR shall satisfy the following requirements: speaker independence, use of continuous speech, about 4000 words in the lexicon, use of linguistic knowledge. The different modules of this system together with a bottom-up control strategy are described in /NIE 85a/. The results generated by various modules and the experiences with the system initiated and strongly effected the considerations of this paper. The main points can be summarized as follows:

- A 100% recognition rate is not achievable at the word level with an acceptable number of word hypothesis.
- The "optimum" with respect to the recognition rate for words, word chains, or complete utterances is not necessarily the "optimum" with respect to the understanding task, if there is no feed back.
- If a grammar for conversational speech exists, it is different from the grammar of the corresponding natural language.
- Even if an utterance is syntactically wrong and/or uncomplete, it is understandable. That means, semantic and pragmatic knowledge is more important than syntactical one.
- Linguistic knowledge can constrain the complexity of the recognition process by giving restrictions and priorities for hypothesis. This capability is not sufficiently used in a bottom-up process.

Therefore, I agree with the following statements given in /GOO 80a/:

"We know that all the available **sources of knowledge must communicate** in the presence of error and uncertainty. ... The problem of **control** in a speech understanding system refers to **how knowledge is organized, activated and focused** to constrain the search. ... The **direction of knowledge flow** is not necessarily from a lower level to a higher level. ... some common representation is necessary if knowledge sources are to interact cooperatively." Contrary to the recognition of spoken texts, cooperation between signal related recognition processes and symbol related understanding processes is necessary to understand spoken utterances in a dialog situation. Therefore, we have to look for system architectures and knowledge representation schemes which **support the integration and flexible cooperation of knowledge and algorithms** developed in both, the speech recognition and the natural language understanding field. Since knowledge and algorithms are well described in other papers of this volume, e.g. /SCH 88a, THU 88a/ this paper will be mainly concerned with the problems of organization, activation, and focusing of the knowledge and the algorithms, in order to develop an architecture for a speech understanding system which fulfils the required properties.

In Section 2 different aspects of knowledge based systems will be discussed with respect to the speech understanding problem. Considering the integration and cooperation of algorithms and different kinds of knowledge, schemes for knowledge representation will be examined in Section 3. Based on the criteria evaluated in both sections **an active homogenuous knowledge base** will be developed in Section 4. This knowledge base is the kernel of the speech understanding system described in Section 5. Some remarks on the computer realization and a first test will conclude the paper.

2. SOME ASPECTS OF KNOWLEDGE BASED SYSTEMS

Generally spoken, the goal of speech understanding is to compute a symbolic description of those contents of a speech signal which are relevant for a given application. Therefore, for information dialogs it is not necessary to interpret the complete signal, but it is sufficient to extract a symbolic description which allows to continue the dialog in an appropriate way. Besides general linguistic

knowledge like acoustic-phonetic, syntax, and semantics, the system has to cover pragmatical knowledge about the discourse domain, knowledge about dialog strategies, and about the former utterances of the actual dialog. The speech understanding task in this situation therefore requires an **explicit representation and an efficient use** of all this **knowledge** by the system. If we try to organize the declarative and procedural (algorithmic) knowledge, we can attack these problems under different viewpoints. First of all, we can seperate different modules from a processing oriented point of view. Contrary, system activities can be grouped around the different kinds of knowledge stored in or created by the system. Both organization principles are independent of speech specific knowledge. The third way is a problem dependent organization. The system is divided into several modules, and each module is related to one level of knowledge like syntax or semantics.

2.1 Organization with Processing Oriented Modules

Fig. 2-1 shows a system architecture which is quite common to most knowledge based systems /NIE 81a/. It is basically built up of the following four modules:

METHODS for doing low level processing like feature extraction, preprocessing, and segmentation. It can also cover more complex algorithms e.g. to calculate distance or similarity measures between a hypothesis and the signal.

KNOWLEDGE to support the understanding process by facts and algorithms (declarative and procedural knowledge).

CONTROL for determining a processing strategy by activating the appropriate algorithms at the proper time using a relevant subset of declarative knowledge and intermediate results.

RESULT DATA BASE for storing the results of processing and making them available to processing algoritms as necessary.

There are several schemes for **knowledge representation** (rules, formal grammars, ATNs, semantic networks, logic), and several **control** strategies and algorithms (backtracking, A^*, heuristics, rules, resolution). The organization of the **result data base** usually depends on the knowledge representation scheme. Some special **methods** may be required by the control algorithm which is used. But in principle,

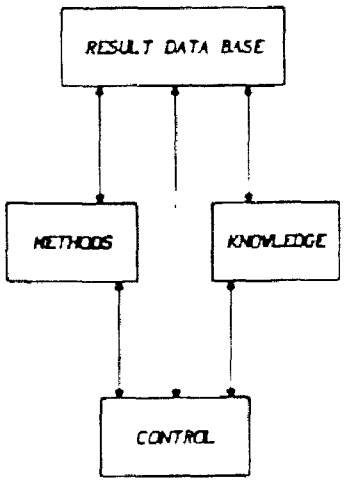

Fig.2-1: Architecture for Pattern Analysis Systems

declarative knowledge	procedural knowledge	a posteriori knowledge
conceptions	inferences	instances
features	extraction, combination	feature values
scheme for judgements	calculation and combination scheme and functions	values
constraints	propagation	restrictions

Fig.2-2: Components of Knowledge

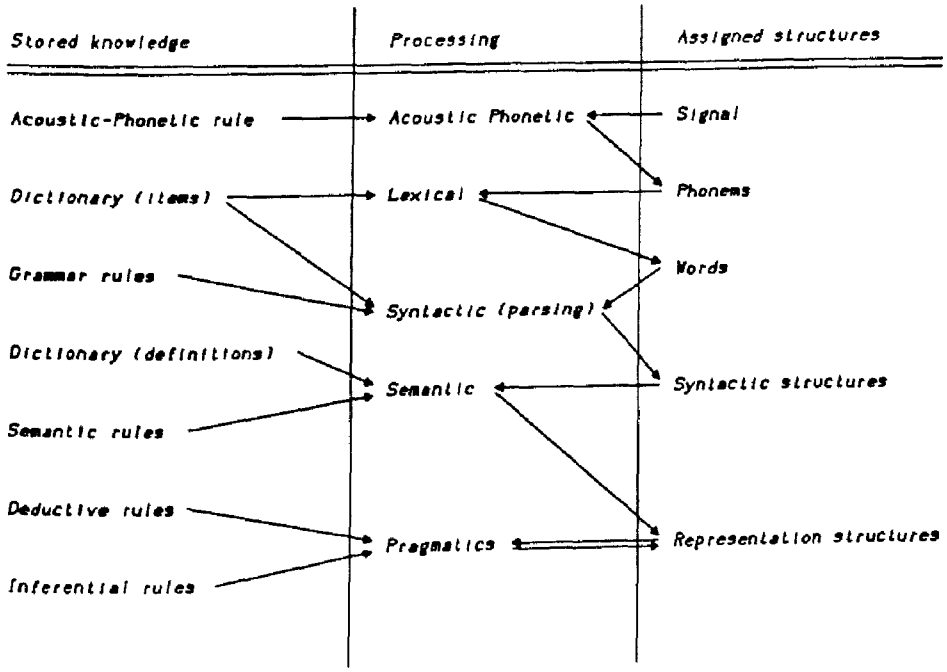

Fig.2-3: Stratified Linguistic Model

the four modules are independent of each other. Nevertheless, the behaviour of the complete system depends on each module and mainly on their cooperative interaction.

2.2 Organization along Different Aspects of Knowledge

In order to achieve descriptions of a speech signal automatically, different aspects of knowledge have to be considered. First of all, there are linguistic **conceptions** like words, syntactical constituents, verb frames, semantic classes etc. Together with task specific conceptions like train schedule information, these conceptions have to be modelled. For an analysis process, **inference processes** are needed to use this knowledge. Based on the a priori known conceptions and on the input signal, **instances** of conceptions are created by these processes. Thus we obtain a triple of associated knowledge aspects: **declarative knowledge** (conceptions), **procedural knowledge** (inferences), and **a posteriori knowledge** (instances). Furthermore, signals have to be transformed into a symbolic description (e.g. words) to initiate the inference processes. For this task **features** are needed. Additionally, if we associate some signal area to one or more symbolic descriptions, the resulting instances have some **properties** which are important for the further processing steps; e.g., the covered sector of the signal. What features and what properties we use, is also declarative knowledge. From the procedural point of view, algorithms are needed to extract and to combine values of such features and properties. Again we get a triple of associated terms, which is summarized in a line of Fig 2-2.

Because signals are noisy, they are a source of errors and ambiguity. One has to take into account that no decision in terms of YES or NO is possible, but every decision can be judged with some kind of certainty or possibility /PRA 85a/. The larger the context of an interpretation is, the more significant a decision is. A calculus for such **judgements** including domain and operations builds up a third triple. Contrary to the composition of facts in order to get more complex interpretations, there are **constraints** which are propagated by such interpretations. E.g., it is a priori known that a noun group can be built up of a noun and adjectives. If additionally the noun is known, the adjectives of the group can be restricted to those ones

which are compatible with the detected noun. Therefore, a constraint can be propagated and this process results in a situation dependend **restriction** on the adjectives.

The described aspects are grouped in Fig.2-2. The a priori knowledge of a system consists of the declarative and procedural components mentioned in this figure. If a module KNOWLEDGE is able to cover and connect these aspects, the module CONTROL is reduced to a monitor, which just activates some processes dependending on the actual situation of the analysis process. Furthermore, most of the METHODS are integrated into the module KNOWLEDGE, and the organization of the RESULT DATA BASE strongly depends on the knowledge representation scheme. Because the system is centered around the module KNOWLEDGE, we extend the demand for "some common representation" (compare section 1.) to the following statement:

> A knowledge representation scheme must cover and connect all the knowledge components shown in Fig.2-2 to support an efficient cooperative interaction between the different components.

The main advantage of this organization principle is, that the communication problem is reduced to the **interactions within the module KNOWLEDGE**. If one knowledge representation language is able to fulfil this statement, most of the problems concerning a cooperative interaction are solved.

2.3 Organization along Levels of Knowledge

A **stratified model of language comprehension** is described in /WIN 83a/. According to this model, the knowledge for speech understanding is made up of levels of procedural and declarative knowledge, that transform the speech signal stepwise into a final interpretation. Fig.2-3 shows such a model with the levels Acoustic-Phonetic, Lexical Retrieval, Syntax, Semantics, and Pragmatics. The advantage of such a model is the modularity in order to get a flexible and expandable system architecture. But in /WIN 83a/ it is pointed out also: "What is needed for effective integration of the levels is a **uniform definition for the structures** that the components accept as input and produce as output", and furthermore that the processing sequence should be separated from the structural levels. The reason for the second point is also given: "Many experiments have demonstrated that

in listening to speech we often **use knowledge of the expected meaning** as an aid in analyzing the sounds in order to decipher the words of the message." There are different possibilities for the required uniform definition for the structures and for a flexible processing sequence.

In Hearsay-II the **blackboard architecture** was introduced. All the modules (also called components or knowledge sources) communicate with each other only via the blackboard. All processing results are written on this structured memory. For the further processing they can be read by other modules. Also, all other kind of communication between modules can only be done via the blackboard. Therefore, the structure of the input and the output of all modules have to be unified. The advantage of this architecture is the modularity. The main disadvantage is that the problem of errors and ambiguity are more or less neglected. There is no explicit CONTROL module, which is able to regulate the activities of the different modules towards a given goal and therefore to manage the search process. Contrary, the **HWIM-architecture** /WOL 80a/ groups the system around a central CONTROL strategy. Only three modules are separated to cover the components of the stratified model. A uniform structure is used to represent the syntactical, the semantic, and the pragmatic knowledge. All these components are integrated into one augmented transition network (ATN) /WOO 70a/. The modules for acoustic-phonetic processing and lexical retrieval are sequentially ordered. The CONTROL module communicates with both the lexical retrieval and the integrated ATN module. Additionally, it is supported by a verification module. This module calculates judgements for hypothesis based on a match between a hypothesis and the signal. The disadvantage here is, that most of the linguistic knowledge is not explicitly stored, because it is not separated from the ATN. E.g., there is no conception for a noun group, for a semantic class, or for a verb frame. Contrary to the stratified model in /WIN 83a/, all such facts are only implicitly used to build up the ATN. The main advantages are the capability of a controlled but flexible processing, the use of uniform and therefore comparable judgements for hypotheses, and the use of predictions based on the ATN.

The two examples show that even system archtitectures for the stratified linguistic model are grouped around one centralized

module: the RESULT DATA BASE at Hearsay-II, respectively the CONTROL module at HWIM. Other potential architectures like a strictly hierarchical or a strictly goal directed architecture are of minor interest /GOO 80a/. Because of the exponential number of data paths, also the complete heterarchical architecture was not realized.

In the previous subsection, the advantages of a system architecture around the module KNOWLEDGE were shown. So far, only requirements concerning different aspects of knowledge were stated for this architecture. The different levels of knowledge were neglected. But this **KNOWLEDGE centered architecture** can also support one main point of the blackboard architecture. If an uniform knowledge representation language is given for all aspects and levels, a uniform μtructure of the RESULT DATA BASE is the consequence. Then the remaining question is, whether one knowledge represenation language is able to integrate at least most of the levels, which are seperated into different modules at Hearsay-II, in an adequate manner. The ATN in the HWIM system does the job of integration. But we have to look for a scheme, which also reflects the different levels of knowledge, in order to keep all the knowledge explicit and to guarantee the modularity of the knowledge base. It was also pointed out, that the judgement calculus should be integrated into the knowledge base. Therefore, the four modules besides the CONTROL strategy of HWIM form the knowledge base, also including the verification. Because of this, the interaction between CONTROL and KNOWLEDGE becomes easier.

3. KNOWLEDGE REPRESENTATION

For a knowledge representation scheme (and language) it was required in section 2, that it should be able

(R1) to integrate the different aspects of knowledge,

(R2) to integrate the different levels of knowledge,

(R3) to make the different levels of knowledge explicit, and

(R4) to build up a modular knowledge base.

Because the intended system strongly depends on the organization of the knowledge base, the question, how adequate is a scheme or language, is of importance. Therefore, adequacy criteria will be

presented, before different knowledge representation schemes are discussed with respect to these **criteria** and to the four **requisitions**, which are summarized above.

3.1 Criteria for Knowledge Representation Languages

In /MCC 69a, BRA 79a, SCH 86a/ the **epistemological adequacy** is accentuated as main criterion for knowledge representation languages. A compact definition for this criterion is given in /MCC 69a/: "A representation is called epistemologically adequate for a person or a machine if it can be used practically to express the facts that one actually has about the aspects of the world." /BRA 79a/ pointed out that an epistemologically adequate scheme must be neutral with respect to a conceptional level of a knowledge base. This level is built up of those conceptions and the relationships between them, which are relevant for a given context of problems. Therefore, a representation scheme should be independent from applications. Further criteria are summarized in /SCH 86a/:

Logical Adequacy: Is the system (language and interpreter) correct, logically complete, and decidable?

Psychological Adequacy: Are the structures used for knowledge representation a model of psychical structures and also the inference processes? Are the errors of the system, comparable to those of a man? Does the system permit contradictions?

Algorithmic Adequacy: Are the algorithms for knowledge utilization of acceptable complexity, and do algorithms exist to monitor the analysis with acceptable complexity?

Ergonomic Adequacy: Is the language manageable and comprehensible?

These criteria do not only ask questions concerning the declarative components of a knowledge representation scheme, but also questions concerning a complete system in the sense of section 2.1 excluding problem specific knowledge. This emphasizes that at least the components pointed out in the first two lines of Fig.2-2 must be judged as one unit with different facets. If a knowledge representation scheme is used to interprete sensor signals one should take a further criteria into account: the handling of uncertain data, and therefore uncertain decisions. In our oppinion it is impossible to extract an unique symbolic description out of speech signals,

take them as being correct, and finally running an understanding process as it is done in systems for natural language understanding. Contrary, it is necessary to describe, how results can be judged in the knowledge representation language itself. Examples for such kinds of knowledge bases are given for example in /DEM 83a, TSO 80a, SAG 85a/. Because of these reasons, the criterion **adequate for sensor data interpretation** is introduced.

It is evident that one system can hardly fulfill all these criteria. Additionally, there is no effective test to decide whether a system realizes these criteria. For the task of pattern understanding psychological adequacy is of less interest compared to the epistemological, the algorithmic, and sensor data adequacy.

If these adequacy criteria are compared with the **requisitions** summarized as R1,..,R4 at the top of this section, it is evident, that both groups have common properties. At one point, those requisitions, which demand the integration of different aspects and levels of knowledge (R1,R2), supply the criterion "epistemological adequacy". Because the "facts ... about the aspects of the world" (see /MCC 69a/) are divided into a number of more special terms, they become more concrete. The most important subcriterion, which results from this division, treats this partial view on epistemological adequacy which concerns the representation of conceptions and their interrelations. If a knowledge representation scheme allows the integration of a judgement calculus (R1) this fact supports the sensor data adequacy. The algorithmic adequacy has references to the procedural aspects of knowledge shown in Fig.2-2. The question on logical adequacy asks for the declarative and the procedural aspects. By the requisition R1 both have to be integrated into one scheme. As a consequence, it is also stated that they are not independent from each other and that they cannot be exchanged individually. Therefore, R1 and the logical adequacy criterion are in correlation. Last not least, the manageableness, which was pointed out as one fact of ergonomic adequacy, is also reflected by the requisitions R1 and R3. Because an integrated knowledge base for speech understanding will be of large complexity, also the manageableness of a large knowledge base in the representation language is an important criterion.

Based on these considerations the following criteria are presented for knowledge representation languages:

(C1) Does the language support the manageableness of large knowledge bases?

(C2) Is the represenation of conceptions and their interrelations epistemologically adequate?

(C3) Does the language support the handling of uncertain data and uncertain decisions?

(C4) Do algorithms exist which monitor analysis processes efficiently?

Together with the requisitions R1,..,R4 these criteria subsume those adequacy criteria which are most relevant for the speech understanding task. Therefore, the requisitions and the criteria C1,..,C4 are used in the following subsection to examine knowledge representation languages.

3.2 Examples and Classification of Knowledge Representation Languages

Following /MYL 83a/ knowledge representation schemes can be classified into the categories **semantic networks**, **logical**, and **procedural schemes**, whereas most knowledge representation languages subsume more than one of these categories. The problems with such a classification are described in /MYL 83a/: "When trying to classify representation schemes we consider the world as a collection of individuals and as a collection of relationships that exist between them. The collection of all individuals and relationships at any time in any one world constitutes a state, and there can be state transformations that cause the creation/destruction of individuals or that can change the relationship among them."

One example to illustrate this is the language PROLOG /KOW 74a/. At one point of view PROLOG is a **logical representation scheme**. Therefore it employs the notations of constant, variable, function, predicate, logical connective and quantifier in order to represent elementary facts. A knowledge base is a collection of terms and formulas. But besides the traditional Tarskian semantics

"B_1 and B_2 and ... and B_m implies A"

the **procedural semantics**

"if you want to establish A,

try to establish B_1 and B_2 and ... and B_m"

is used. This procedural point of view establishes how knowledge can

be used and is therefore comparable to calculus for proving theorems, like the Gentzen calculus or the resolution method /MAN 74a/. Because the use of knowledge is the crucial point for the procedural semantics, we prefer the notation that a **pragmatics** is defined for **the language**, respectively for logical schemes.

Procedural schemes view a knowledge base as a collection of active processes and agents. Most procedural schemes have been influenced by LISP and LISP itself was a favourite represenation language. Schemes like **production systems** /HAY 82a/ and PLANNER /HEW 71a/ offer activation mechanisms for processes. In both schemes a knowledge base is built up of pairs. Each pair consist of a pattern and one or more actions which manipulate the result data base. If the pattern of a pair can be successfully matched to the data base the corresponding theorem in PLANNER respectively the action of the rule is executed. One difference is, that a theorem can directly call another one, while the action of a rule can only modify the result data base. The PLANNER control module uses the backtracking algorithm. This is also used in many applications based on production systems. Nevertheless, numerous other algorithms are used to monitor such systems. Contrary, the ACTOR formalism /HEW 77a/ places the control structures into the foreground. All objects of a knowledge base are viewed as active agents, called actors. They have the capability to send and to receive messages. The messages themselves are actors. An object is specified by the kinds of messages it can receive and all the actions, including the sending of messages, it takes. The actions depend on the message the object received. One of the main points of **semantic networks** is the information retrieval. Although the languages based on such networks are of large diversity, there exists a most basic form. A priori and a posteriori knowledge is expressed by nodes and directed labeled links. The former stand for conceptions or classes of conceptions (mainly a priori) and individuals (mainly a posteriori), the latter for binary relations over these. The main problem of the scheme was that most of the early languages had little or no semantics for the types of nodes and links they used. The nessecity for a semantics and an epistemological adequacy was pointed out in /WOO 75a, BRA 79a/. In order to define a unique interpretation of the different types of nodes and links in a language, this set must be restricted. On the other hand, to get an

epistemological adequate language, this set must be sufficient to build up knowledge bases for all possible or at least a large number of applications. Most network languages offer different organizational axes for structuring a knowledge base /FIN 79a/. The most common axes are:

Classification: A real world object is associated with its generic type(s): This axis forces a distinction between a concept, which is the intensional description or a prototype of a conception, and an instance, which is a member of the extensional set of a concept.

Aggregation: A concept or an instance is related to other concepts or instances respectively, which describe their components or parts.

Generalization: It relates a concept to more generic ones. Generalization, often called is_a, defines a hierarchy in the network due to a partial order. In most approaches properties associated with a general concept are inherited to the more special ones, unless they are not explicitely modified.

Semantic network languages like KL-ONE /BRA 85a/ and PSN /LEV 79a/ are influenced by the notation of **frames** /MIN 75a/. Such frames are complex data structures for representing stereotypical informations for a field of problems. A frame has slots for the objects which play a role in the situation as well as conditions between the slots. Furthermore, processes and facts are attached to the slots of a frame, like "if added", "if needed". In KL-ONE and PSN frame like data structures are used to build up the nodes of the semantic network. Links are described inside such a data structure by slots of different types, one for each link type. The interpretation of the slots and their facets is defined with respect to the semantics of the semantic network. While KL-ONE only uses procedural attachment associated with the slots, in PSN also a pragmatics (procedural semantics) is defined. Besides others, procedures to build up or to delete instances are associated with each concept. The slots in a concept are divided into prerequisites and consequences with respect to the instantiation procedure of the concept.

3.3 Discussion of Knowledge Representation Languages

For each class of knowledge representation schemes one language is choosen for the following discussion:

PROLOG for logical schemes

EMYCIN /BUC 85a/ for rule based system shells

PSN for semantic networks

As mentioned above the inference process of PROLOG is given by the definition of the procedural semantics. Therefore, PROLOG covers declarative and procedural aspects (R1). Vice versa, the rules in EMYCIN also allow a declarative description of facts. But both do not integrate the handling of numerical features (R1). Because it is a logical language, PROLOG offers no help to operate with uncertain data and uncertain decisions (C3). On the contrary, EMYCIN allows scores for rules. The main problem at this point is that both, PROLOG and EMYCIN, have to decide whether the premise of a rule is true or false. As a consequence, if uncertain data and judgements on interpretations are used, there is a need for thresholds. Because rules and formulas can be formulated independent of each other, both languages are modular (R4). They can also integrate different levels of knowledge (R2). But these facts imply that such levels cannot be made explicit (R3). Whereas identical conceptions are identified in PROLOG and EMYCIN implicitly by utilization of an identical name, each conception is representad exactly once in a semantic network. All relationships, the conception is associated with, are centered in the node standing for the conception. Therefore, semantic networks are more suited for large knowledge bases (C1). But this kind of organization is also helpful to keep a knowledge base modular (R4) in the following sense: If one conception has to be changed, it is known in a semantic network, where and how often it occurs in the knowledge base. Applications of PSN, e.g. /TSO 80a/, show that it is possible to integrate different aspects and levels of knowledge into one knowledge base (R1,R2), and that PSN supports the handling of uncertain data (C3). Contrary to PROLOG, EMYCIN and PSN do not include a complete monitor. But as it is shown by examples, there exist monitors which satisfy criterion C4. By their built in pragmatics, all these three languages prefer one direction, goal-directed or data-driven, for analysis processes. Thus, a flexible propagation of constraints (R1) and flexibility of processing sequences are limited.

The **epistemological adequacy** (C2) is one of the most serious problems concerning a knowledge representation language. For semantic

networks, this adequacy strongly depend on the built in links and the distinction of different types of properties /BRA 79a/. The set of epistemological primitives is defined in KL-ONE and PSN by the link-types, which were presented in the last subsection, **attributes**, which describe further properties and relationships, and **structural relations** to describe and test relations, which must hold between referred conceptions and attributes. In PROLOG and EMYCIN it is not possible to distinguish such different classes of relationships and properties in an explicit way. All of them and the conceptions are represented uniformly by predicates and formulas. Additionally, there is no way to describe an inheritence between a class of conceptions and a corresponding superclass. In order to motivate the need for an additional link type in a semantic network language, we present an example given by /MYL 83a/: "for example the parts of John Smith, viewed as a physical object are his head, arms etc. When viewing as a social object, they are its address, social insurance number, etc." Two "worlds" or **conceptional systems** are distinguished in this example. A concept modelling a person has different parts within each of these systems. Parts in the social system are social conceptions, parts in the physical system are physical conceptions. In speech understanding conceptional systems like syntax, semantics, pragmatics build up the knowledge base. Therefore, links between conceptions have to be distinguished, whether they connect conceptions inside one or conceptions belonging to different conceptional systems. In /SCH 86a/ it is pointed out, that within one conceptional system parts of concepts may exist, which can only be described adequately inside the context of those conceptions they are part from. As an example, social structures like family, father, mother, and married couple are given. It is worked out, that an adequate description of these conceptions requires a definition by sets, and that a definition by atoms is not adequate. A **context dependent** definition like husband in /SCH 86a/

husband(x) := u (u=<x,y> married_couple(u))

could be established by defining "husband" to be an aggregation of "married couple" but also to depend on the context of "married couple".

4. A HOMOGENUOUS KNOWLEDGE BASE FOR SPEECH UNDERSTANDING

After the discussion of knowledge representation languages we now turn to the presentation of a particular semantic network language. this language is the kernel of the system shell ERNEST /KUM 87a/. There is a clear distinction between the syntax, semantics, and pragmatics of this language. In order to explain the interpretations, i.e. the semantics, of the ERNEST structures, the knowledge base designed for the system EVAR will also be presented in this section. The knowledge base is called homogenuous, because it integrates the different aspects and levels of knowledge discussed above.

4.1 The Declarative Aspects of the Knowledge Base

First of all the ERNEST network language distinguishes three types of nodes. A **concept** represent the model of a conception. According to classification (see 3.2), **instances** are associated with concepts. The connection between instance nodes, concept nodes and signal areas is shown in Fig.4-1. In this figure the three word hypothesis "train", "time", and "Hamburg" stand for the signal. Three concepts "noun", "source", and "goal" are given. Instances are denoted by circles. Each instance establishes a connection between exactly one concept and an unique signal area (word hypothesis). Such an area can be

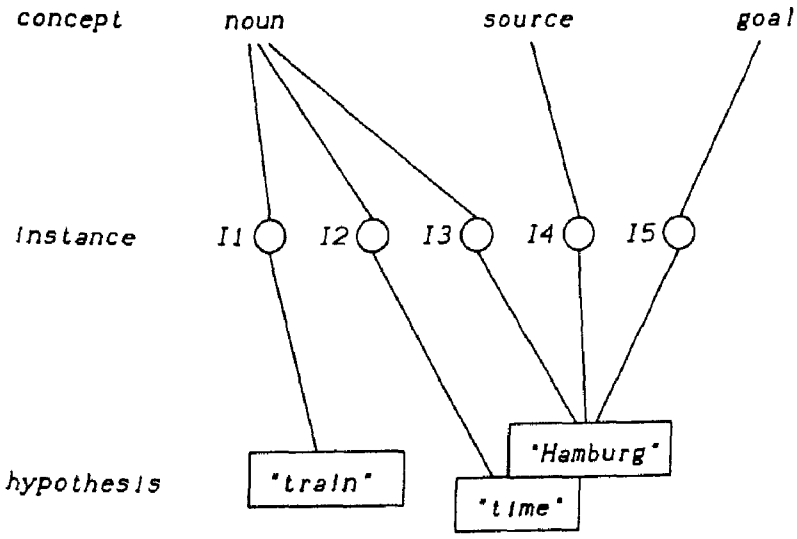

Fig.4-1: Relations between Concepts, Instances and Hypothesis

connected to more than one concept via instances and one concept to more than one area. It is obvious, that there are instances which compete each other: I4 and I5 because "source" and "goal" is a contradiction, and I2 and I3 because the signal areas overlap. Contrary, I3 and I4 are compatible, because "noun" and "source" belong to the different levels (conceptional systems) syntax respectively semantics. As illustrated by its name **modified concept**, the third node type is similar to a concept, but it is restricted with respect to a given situation of an analysis process. For example, in German adjectives and nouns have to agree in genus, number, and casus inside a noun group. A priori these attributes are unknown for the group or one member. If an adjective is yet instantiated, modified concepts for noun group and noun can be built up. This is done by restricting the attributes mentioned above with respect to the detected adjective. With the interpretation of the three node types also the link types **instance** and **instance of** are fixed according to the links between concepts and instances in Fig.4-1.

Similar to PSN, all node types are syntactically described by frame like structures. Fig.4-2 shows those parts of the data structures building up an ERNEST concept, which are relevant for the speech understanding application. Besides internal properties, the structures subsume all the links to and from other concepts. The structures for modified concepts and for instances differ only slightly from those for concepts. Therefore, the interpretation of the structure "concept" the substructures and the items, which together build up a concepts, are given in the following. If there are substantial differences to the other node types, it will be mentioned.

Besides the instance relationship four organizational axes are distinguished. All of them define a nonreflexive partial order on the set of concepts. Trees of concepts are constructed by the link **specialization**. This link type establishes the inverse of the generalization as described in section 3.2. All substructures of the groups part, concrete, attribute, analysis attribute, structural relation, and analysis relation referred in a concept are inhereted to its specialization, unless they are explicitely modified by marking this fact in the item **modified** of the superimposing

concept

name of concept	→	text
degree	→	4 integer
priorities	→	5 integer
information	→	text
specialization of	→	link to concept
specialization	→	list of links to concepts
context of	→	list of concepts
part of	→	list of links to concepts
part	→	list of link descriptions
modality	→	list of modality descriptions
attribute	→	list of attribute descriptions
local attribute	→	liste of attribute descriptions
structural relation	→	list of relations
concrete of	→	list of links to concepts
concrete	→	list of link descriptions
analysis parameter	→	list of attribute descriptions
analysis relation	→	list of relations
identification	→	list of identifications
judgement	→	procedure
instance	→	list of instances

modality description

obligatory	→	list of roles
optional	→	list of roles
inherent	→	list of roles
adjacency	→	adjacency
coherent	→	YES or NO

value

type	→	type
meaning	→	meaning
value	→	value array

procedure

name	→	text
argument	→	list of arguments
inverse procedure	→	name of procedure

adjacency

dimension	→	integer
roles	→	list of roles
diagonal	→	integer vector
matrix	→	bitmatrix

attribute description

role	→	text
domain	→	type
selection	→	range
modifies	→	YES or NO or role
dimension	→	2x2 integer
computation of value	→	procedure
adjacency dependent	→	YES or NO
restriction	→	procedure
preference	→	list of ranges
active graphic	→	procedure

link description

role	→	text
domain	→	list of concepts
context depending	→	YES or NO
modifies	→	YES or NO or role
dimension	→	2 integer
restriction	→	procedure
transformation	→	value
preference	→	list of ranges

relation

role	→	text
test of relation	→	procedure
adjacency dependent	→	YES or NO

range

kind	→	kind
type of values	→	type
complement	→	YES or NO
values	→	list of values

identification

path1	→	list of roles
path2	→	list of roles

Fig.4-2: Data Structure Building up a Concept in ERNEST

substructure in the specialized concept. Aggregation is notified by the link **part**. Part and specialization relationships are restricted in the way, that they are only allowed inside the same conceptional system. If a concept is referred by the inverse link **part of** and in the item **context of**, this indicates that the concept itself is context dependent from that twice referred concept. (see 3.2). Links between different conceptional systems are established by the **concrete** relationship.

Fig.4-3 shows a snapshot on the network of the knowledge base. In order to simplify the following explanations, each concept in this figure has a prefix, which illustrates its membership to one of the conceptional systems **pragmatics** (P), **pragmatic classification** (PC), **semantics** (S), **semantic classes** (SC), or **syntax** (SY). At the very bottom, the concept WORD build up one interface of this network to the signal oriented analysis modules /SCH 87a/. At the top level, pragmatical conceptions like P_TRAIN_CONNECTION, with parts P_VF_FAHREN or P_FROM_TIME are shown. They build up the application oriented conceptions. E.g., P_VF_FAHREN relates the corresponding verbframe for "fahren" (to go, to drive) to the application. The concept S_VF_FAHREN itself is indepedent from the application. From the applications point of view, to ask for a train connection using the verb "fahren" requires time information (P_FROM_TIME). Contrary, S_TIME is not referred by S_VF_FAHREN. Concepts of the conceptional system describing the pragmatic level are connected via concrete links to concepts of the PC and the S levels. E.g., P_VF_FAHREN is concretized by S_VF_FAHREN, and P_FROM_TIME by both PC_TIME_INTERVAL and S_TIME. From the concept S_VF_FAHREN, deep cases like S_INSTRUMENT, or S_SOURCE are referred. They are concretized by both semantic classes and syntactical constituents. Syntactical conceptions like nominal phrase complex (SY_NPC) have other syntactical concepts like preposition phrase (SY_PP) as parts, which itself is built up of SY_NP (nominal phrase) and SY_PREP (preposition). While Fig.4-3 was restricted to a few concepts and links, Fig.4-4 gives a condensed view on the complete network. Here the part links are omitted. They connect only concepts, which are members of one conceptional system. Arrows between blocks and/or single concepts (capital letters) stand for bundles of links of the desired type. The membership of a block or a concept to a

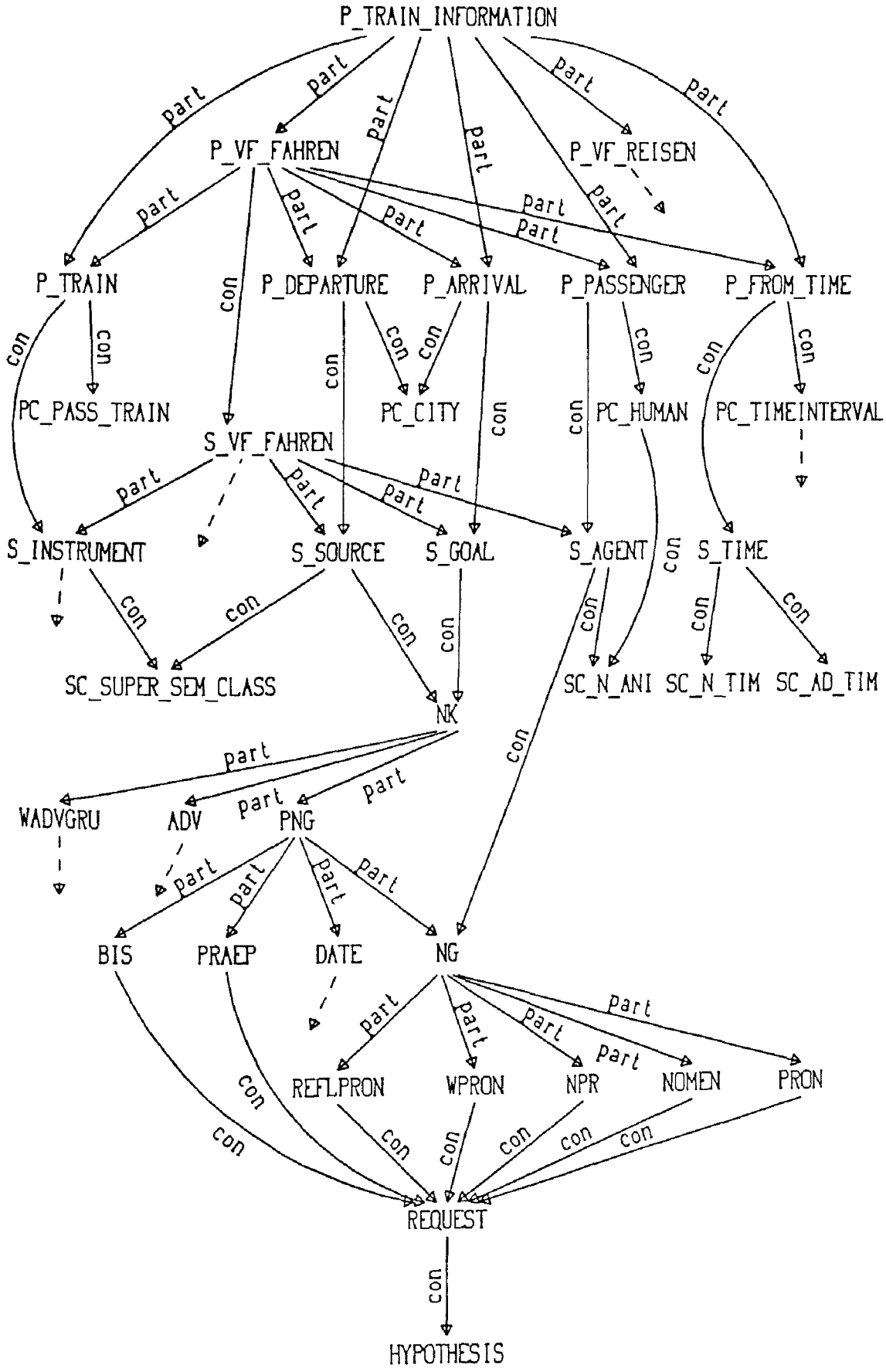

Fig.4-3: Snapshot on the Knowledge Base

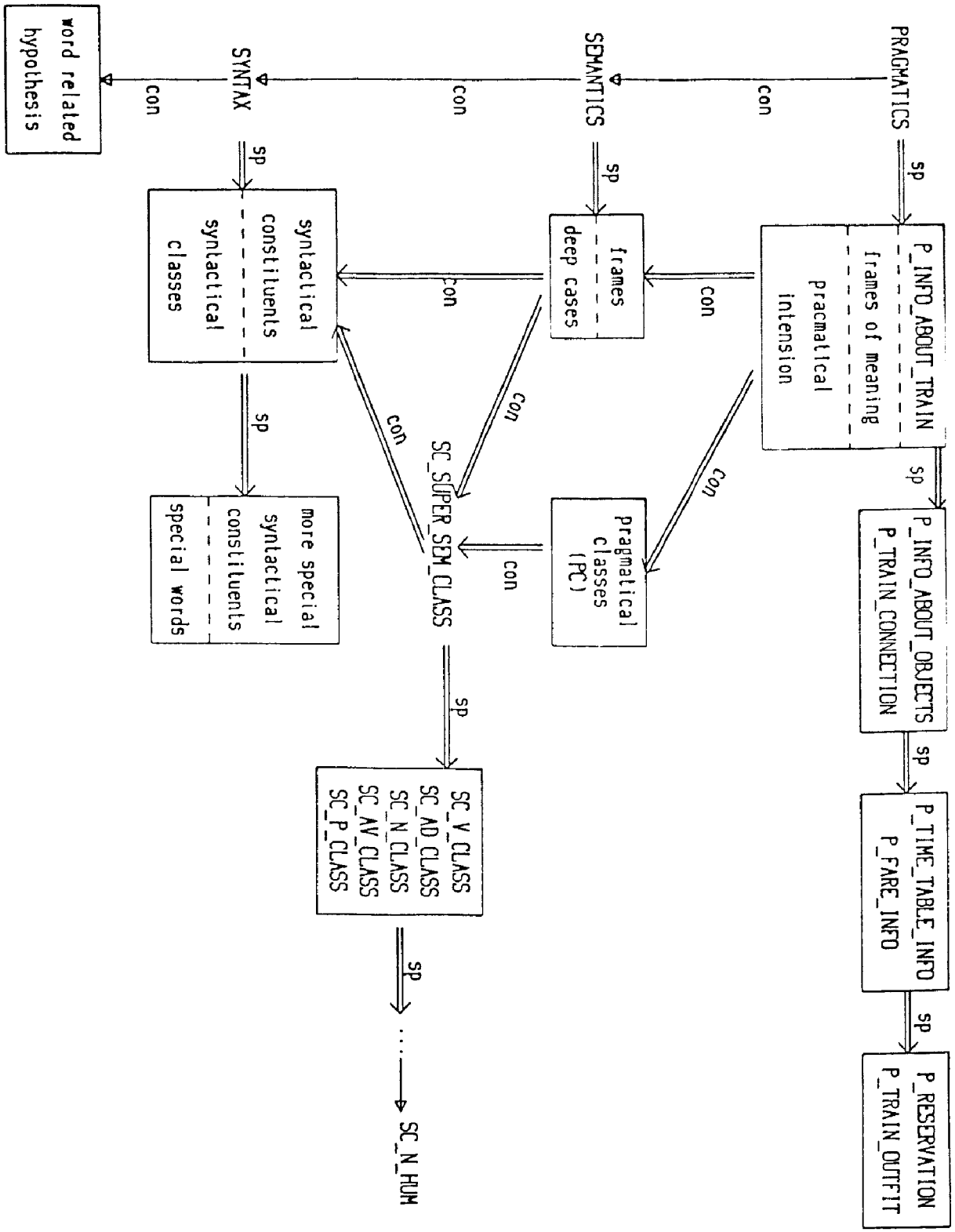

Fig.4-4: Condensed View on the Knowledge Base

conceptional system is denoted by the prefixis also used in Fig.4-3.

The complete definition of each part and concrete relationship is done by the substructure **link description**. Different from most similar knowledge representation languages, the **dimension** item inside a link description causes no differentiation for the correponding **role** in the instances. However, the role in an instance has to be filled with just as much instances the dimension intervall requests. Parts and concretes defined in a concept or inherited to the concept may be grouped into different **modality descriptions**. Each such substructure establishes one admissable combination of **obligatory**, **optional**, and **inherent** links, whereas one link can be inherent and optional or inherent and obligatory in the same modality. The substructure is completed by an **adjacency description** which allows a compact notation of required time neighbourhoods of parts. If no gaps are allowed between adjacent parts, this fact is notified by the term YES in the item **coherent**. An example of modality and adjacency descriptions is given in Fig.4-5. Fig.4-5a shows the graphical ATN representation of the concept SY_NP of Fig.4-3. This results in the modality description in Fig.4-5b. For simplicity, the roles of the parts are represented by the name of the corresponding concept but written in small letters. The adjacency description, i.e. the matrix, for first modality set of SY_NP is shown in Fig.4-5c. Because it is possible to describe the facts of a modality description also by specialized concepts and the adjacency in the substructures relation, both are not epistemological primitive. But they give compacter knowledge bases and therefore aids the ergonomic adequacy.

The intensional description of a concept is completed by its **attributes** and **structural relations**. E.g., the CNG features of nouns, adjectives, noun phrase are attributes. That they shall agree inside one noun phrase are structural relations. The main items in the substructures defining an attribute description respectively a relation are the **computation of value** for an attribute and the **test of relation** when building up an instance. A **procedure** defintion in the ERNEST-network includes the explicit notation of the **arguments**, and it is possible to refer also the inverse of the procedure. While the activation of the procedure itself results in concrete values if all arguments are known, the inverse is able to restrict the domain set, especially the **selection** of attributes in modified concepts. In

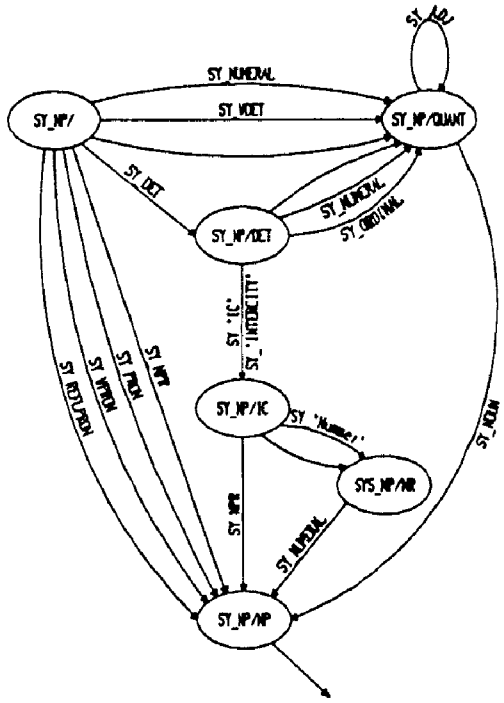

Fig.4-5a: Graphical Representation of an ATN

```
MODALITY_DESCRIPTIONS:

   MODALITY:
      OBLIGATORY: noun
      OPTIONAL:   det, wdet, numeral, ordinal, adj
   MODALITY:
      OBLIGATORY: ic, npr
      OPTIONAL:   ---
   MODALITY:
      OBLIGATORY: intercity, npr
      OPTIONAL:   ---
   MODALITY:
      OBLIGATORY: ic, numeral
      OPTIONAL:   nummer
   MODALITY:
      OBLIGATORY: intercity, numeral
      OPTIONAL:   nummer
   MODALITY:
      OBLIGATORY: npr
      OPTIONAL:   ---
   MODALITY:
      OBLIGATORY: pron
      OPTIONAL:   ---
   MODALITY:
      OBLIGATORY: wpron
      OPTIONAL:   ---
   MODALITY:
      OBLIGATORY: reflpron
      OPTIONAL:   ---
```

Fig.4-5b: Modality Description

	noun	det	wdet	numeral	ordinal	adj
noun	0	1	1	1	1	1
det	0	0	0	1	1	0
wdet	0	0	0	0	0	0
numeral	0	1	0	0	0	0
ordinal	0	1	0	0	0	0
adj	1	1	1	1	1	4

Fig.4-5c: Adjacency Description

a similar way the selection of an attribute may be restricted, if not all the argument values are known. The complete domain set of an attribute is described by the three items **domain**, **selection**, and **dimension**. The domain item refer the type, e.g. integer or casus. By the selection this type can be constraint to certain valus, e.g. to the numbers 1 to 10 or to the subset nominative, accusative. The dimension item gives the number of values which are necessary to fill the property in an instance, e.g. at least 1 at most 10. Arguments are described by a role or a pair of roles, respectively. If only one role is referred, the argument is taken from the instance or modified concept itself. By two roles attributes of parts or concretes are referred to by the role in the corresponding link description and the role of an attribut in this part or concrete. As þdditional argument the adjacency matrix is allowed. This is notified by the content of the item **ajancency dependent**. While the attributes show the features of a concept, **analysis attributes** accept quantitative parameters, which are necessary for the analysis process but do not contribute the intensional description. One example of such an attribute, is a data structure which is called "mask". It is analysis attribute of each concept of the network (see 4.2). Quite the same situation differs structural relations from **analysis relations**. All the attribute and relation groups described so fare are implicitely inhereted to more special concepts. Contrary, **local attributes** are not inhereted, they describe local properties.

The function to judge an instance with respect to the conception modelled by the corresponding concept is referred in the item **judgement** of the concept. Arguments of this function are the judgements of the parts and concretes, the restrictions of link and attribute descriptions, and the results of both kinds of relations. All arguments are ·eferred by the role of the corresponding substructure. The calculus for judgements is not fixed by the ERNEST network language. In section 5 the use of judgement vectors, which reflect quality, certainty, and priority of a (modified) concept, or an instance, for speech understanding will be discussed.

4.2 Procedural Aspects of the Knowledge Base

Besides the procedures and their inverse, which are attached to

substructures of a concept, a **pragmatics** is defined for the ERNEST network language. Different from PSN and analogous to PROLOG, this procedural semantics is independent from the content of a knowledge base. It is build up of **five rules**, which only look for concepts, modified concepts, and instances in the premise. Starting with a few goal concepts, recursive applications of these rules results in a search tree, which is the skeleton of the search tree of the analysis process. Out of the skeleton, the search tree is generated because of procedures, which do not produce an unique result but competing values. For example, one noun may result in more than one instance of the concept noun, because there are different values for the attribute casus, which compete each other.

In order to illustrate the five rules, an example of an analysis process is chosen in Fig.4-6a. It is assumed that up to this situation the instances are yet created. Three of them are shown in the figure. They are denoted by $in_i(A)$, where A is a concept and i the index for the instance. The words, which are assumed to be the major content for the instances are also shown. Additionally, the instances are placed at those positions, which are established by the corresponding concepts. The snapshot on the knowledge is chosen with respect to Fig.4-4.

The most basic rule is

RULE 1: IF **for a concept A or a modified concept mod(A)** with respect to one obligatory set of a modality of A, **instances** for those concepts **exist**, which are referred to by the following slots in A or slots inherited to A without modification:

- **concrete** AND
- **part**, if the item **context dependent** in the link description is equal to NO
- one of **context of**, if this slot is not equal to NO

THEN build up a **partial instance** $inp_j(A)$ as follows

- construct an empty instance of A
- connect the instance with those referred to by the premise
- activate the attached procedures in the sequence: restrictions of links, calculation of attributes, restrictions of attributes, tests of relations, judgement

Because of context depending parts and optional links, values do not exist for all arguments of the activated procedures. Nevertheless the

knowledge of existing values of arguments and the procedure itself can be used, if the following strategy is applied. The selection values are also transferred to the procedures. The procedures themselves decide whether the existing values and the selections are sufficient for the **estimation of results**. In the case of attributes, this estimation is a new, more restrictive selection. For the other cases, which all are tests, the estimation must be optistic. Except the creation of an empty instance the same action takes place for the second and the third rule.

RULE 2: IF a **partial instance** $\mathbf{inp_j(A)}$ of a concept A exists AND
instances for all those concepts exist, which are refferd to as **part** with item **context depending** is equal to **YES** in the link description and are members of the **obligatory** set which was used for the construction of $inp_j(A)$
THEN build up new **instances** $\mathbf{in_k(A)}$ out of $inp_j(A)$

RULE 3: IF an **instance** $\mathbf{in_j(A)}$ of a concept A exist AND
at least one instance of a concept exist, which is **optional** due to the modality used for constructing $in_j(A)$
THEN build up **extended instances** $\mathbf{in_k(A)}$ out of $in_j(A)$

In the situation of Fig.4-6a, the premise of RULE1 is satisfied for the concept S_SOURCE. Because no context dependent parts are referred, RULE2 can be applied immediatelly after RULE1. The result is the instance iv_m(S_SOURCE) which is partially shown in Fig.4-6b together with the most relevant definitions of the concept S_SOURCE. The analysis attribute "mask" is a vector, where every position stands for one frame of the input signal. By "-" free positions are marked. The "x"-es represent positions of words which build up the instance.

The information represented by the instances iv_m(S_SOURCE) and in_l(VG) is able to restrict the concept S_VF_FAHREN. Based on such a modification it would be also possible to restrict the concepts S_OBJECT and/or S_GOAL. For the modifications of concepts the following rules are introduced:

RULE 4: IF for a **concept A** or a **modified concept** $\mathbf{mod_j(A)}$,
a new modified concept or a new instance were created for a concept, which is referred to as
part, or **concrete**, or **context_of**
by the concept A

THEN create a new **modified concept mod_k(A)** out of A or mod_j(A), respectively, as follows
- construct an empty modified concept of A
- connect this modified with all instances referred to by the premise and those, which are already referred to by mod_j(A)
- activate the procedures like in RULE1

RULE 5: IF for a **concept A** or a **modified concept mod_j(A)**,
a new modified concept or a new partial instance were created for a concept B, which is referred to as
part_of, or **concrete_of**
by the concept A

THEN create a new **modified concept mod_k(A)** out of A or mod_j(A), respectively, as follows
- construct an empty modified concept of A
- connect this modified with all instances referred to by the premise and those, which are already referred to by mod_j(A)
- activate the following procedures in the sequence:
  inverse restriction of link descriptions of B, inverse calculation of attributes of B, inverse tests of relations of B, restictions of limks of A, calculation of attributes of A, restrictions of attributes of A, tests of relations of A, judgement of A

The premise of RULE4 is satisfied for the concept S_VF_FAHREN. The differences between this concept and the resulting modified concept mod_i(S_VF_FAHREN) are illustrated by Fig.4-6c. Notice, that no value but only a new selection is calculated for the attribute "mask". With mod_i(S_VF_FAHREN) the premise of RULE5 holds for the concept S_GOAL. The modified concept mod_n(S_GOAL) can be created, Fig.4-6d. Besides the new selection for "mask", the domain of the two concret descriptions is restricted. The verbframe "fahren" requires special semantic classes for S_GOAL and a special syntactic realization. These are notified as restrictions in the corresponding link descriptions in the concept S_VF_FAHREN. The inverse procedures of these restrictions modify the corresponding domains in mod_n(S_GOAL).

The first three rules are suffient to create and to extend instances with respect to optional links. RULE4 and RUL5 make it

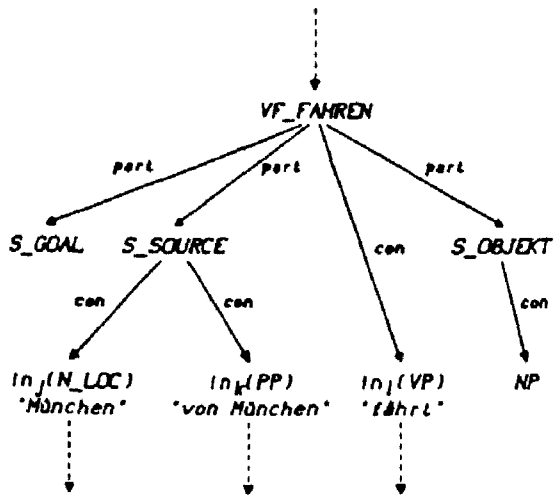

Fig.4-6a: Snapshot of a Search Graph Node

```
S_SOURCE:

   CONCRETE: sem_class
      DOMAIN:N_LOC
   CONCRETE: synt_realization
      DOMAIN:PP
   ANALYSESPARAMETER: mask
      ARGUMENT:  sem_class.mask,
                 synt_realization.mask
      SELECTION: [--------------------------]
      VALUE:     [--------------------------]
   RELATION: partial_identity
      ARGUMENT: sem_class.mask,
                synt_realization.mask
      VALUE:    ----


in_m(S_SOURCE)

   CONCRETE: sem_class
      DOMAIN:in_j(N_LOC)
   CONCRETE: synt_realization
      DOMAIN:in_k(PP)
   ANALYSESPARAMETER: mask
      ARGUMENT:  sem_class.mask,
                 synt_realization.mask
      SELECTION: [-------------xxxxxxx------]
      VALUE:     [-------------xxxxxxx------]
   RELATION: partial_identity
      ARGUMENT: sem_class.mask,
                synt_realization.mask
      VALUE:    0.9
```

Fig.4-6b: The Concept S_SOURCE and an Instance

```
S_VF_FAHREN

   PART: source
      DOMAIN:S_SOURCE
   PART: goal
      DOMAIN:S_GOAL
   PART: object
      DOMAIN:S_OBJECT
   CONCRETE: verbal_phrase
      DOMAIN:VP
   ANALYSESPARAMETER: mask
      ARGUMENT:  goal.mask,
                 source.mask,
                 object.mask,
                 verbal_phrase.mask
      SELECTION: [--------------------------]
      VALUE:     [--------------------------]
   RELATION: adjacency
      ARGUMENT: goal.mask,
                source.mask

mod_i(S_VF_FAHREN)

   PART: source
      DOMAIN: in_m(S_SOURCE)
   PART: verbal_phrase
      DOMAIN: in_l(VP)
   ANALYSESPARAMETER: mask
      ARGUMENT:  goal.mask,
                 source.mask,
                 object.mask,
                 verbal_phrase.mask
      SELECTION: [xxxxx---------xxxxxxx------]
      VALUE:     [--------------------------]
   RELATION: adjacency
      ARGUMENT: goal.mask,
                source.mask
      VALUE:    ---
```

Fig.4-6c: The Concept S_VF_FAHREN and a Modified Concept

```
S_GOAL:

   CONCRETE: sem_class
      DOMAIN:SUPER_SEM_CLASS
   CONCRETE: synt_realization
      DOMAIN:PP, NP
   ANALYSESPARAMETER: mask
      ARGUMENT:  sem_class.mask
                 synt_realization.mask
      SELECTION: [--------------------------]
      VALUE:     [--------------------------]
   RELATION: partial_identity
      ARGUMENT: sem_class.mask
                synt_realization.mask
      VALUE:    ---

mod_n(S_GOAL)

   CONCRETE: sem_class
      DOMAIN:N_LOC, A_LOC
   CONCRETE: synt_realization
      DOMAIN:PP
   ANALYSESPARAMETER: mask
      ARGUMENT: sem_class.mask
                synt_realization.mask
      SELEKTION: [####o---------o#####o------]
      WERT:      [--------------------------]
   RELATION: partial_identity
      ARGUMENT: sem_class.mask
                synt_realization.mask
      VALUE:    ---
```

Fig.4-6d: The Concept S_GOAL and a Modified Concept

feasible to adjust and therefore to restrict a complete knowledge base to every new instance by creating modified concepts. RULE5 can be applied recursively until concepts which have an empty premise with respect to RULE1. These concepts are exactly those which build up the interface between the declarative part of the knowledge base and the module which generates word hypotheses.

5. ANALYSIS STRATEGY AND JUDGEMENTS

In the last subsection the inference rules were presented. Their recursive application build up the skeleton of the search space of the analysis process. Competing word or word chain hypotheses together with competing intermediate results split up this skeleton into the complete search space. The A*-Algorithm /NIL 82a/ in combination with judgement vectors, which scores both instances and search space nodes, is used to guide the analysis process. This process itself can be divided into the following major subprocesses. First of all, based on a few word hypotheses so called "goal concepts" are selected out of the knowledge base. For these concepts it is required, that they are located at the conceptional sub-system "pragmatics" of the knowledge base. Each of the selected word hypotheses forms a successor of the start node in the search space. By the second subprocess, nodes of this space have to be judged with respect to the required properties of judgements for the A*-Algorithm. We use judgement functions, which are based on the content of the node. This subsumes all the instances and modified concepts which were created on the path from the start node to the node to be judged and the a priori knowledge base. The third kind of subprocesses expands search space nodes by applying rules of the rule set, which was described in the last subsection. Depending on the applied rule and the underlying concept, one or more new instances or modified concepts are created. If the application of a rule results in exactly one new modified concept, no successor is generated for the actual search space node. In all other cases one successor node is constructed for each created instance or modified concept. All the three subprocess types use judgements on search space nodes, instances, modified concepts, or concepts. Before we give an outline

of the complete analysis process, the different scoring values which build up the judgements of the different objects are explained.

5.1 Knowledge Based Priority Scores

A priori for each concept of the network the **concept priority vector** is defined. These vectors can be calculated automatically only based on the different links of the network but independent of the special content of the concepts. They give answers to the following questions:

- What is the maximal distance via concrete links to the very bottom level? Concepts which have such a maximal distance belong to the conceptional system "pragmatics", of the network. ($prio_1$)
- What is the minimal distance via concrete links to one concept at the very bottom level, i.e. the conceptional system "word related hypothesis" , of the network ? ($prio_2$)
- What is the minimal distance via concrete and/or part links to one concept at the very bottom level of the network ? ($prio_3$)
- How special is the concept ? ($prio_4$)
- How complex is the concept with respect to the part relationship ($prio_5$)

A concept A is of higher priority compared to a concept B, if in lexical order for bottom-up rules the vector ($prio_1$,$prio_2$,$prio_3$,$prio_4$,$prio_5$) of A is greater than that of B and for top-down rules the vector (-$prio_2$,-$prio_3$,$prio_4$,-$prio_5$) of A is greater than that of B. Modified concepts inherit their priority vector from the corresponding concept.

Besides this static priority measurement two dynamic priority scores are defined for modified concepts and for instances. The first one is called the **pragmatic priority** pP /EHR 87a/. For many words in the dictionary a possible set of pragmatic conceptions can be determined. With this property for each word a pragmatic bit vector pbv(w) is defined. Each bit represents a conception. The conceptions themselves are organized in a tree structure with the most general one as root. The pragmatic bitvector is defined as an attribute or an analysis attribute in each concept of the network. For a word w the pragmatic bitvector has "1" for those conceptions that could be realized by the word or that can realize the domain of a part. If the

word is a pronoun or a determiner, it gets the "1" with respect to its semantic class. For this there exist a mapping function from semantic classes to pragmatic conceptions. The **pragmatic bit vector of a group of words** $w_1,\ldots,w_n$ is defined by

$$pbv(w_1 \ldots w_n) := pbv(w_1) \text{ AND } \ldots \text{ AND } pbv(w_n)$$

The **pragmatic priority** $pP(w_1 \ldots w_n)$ is defined as the number of "1" in the pragmatic bitvector and has the following properties:

- If the pragmatic priority of a group of words is equal to 0, then the group is pragmatically inconsistent.
- The smaller the priority the better the hypothesis with these words

For instances and modified concepts the pragmatic bitvector is calculated respectively estimated on the basis of the underlying word hypotheses and the selections of the attribute "pragmatic bitvector" which are a priori notified in the concepts.

Additionally, also for modified concepts and instances a **semantic priority** /EHR 87a/ is defined. This priority measurement is only calculated for modified concepts and instances which represent verb, noun or adjective frames. But it is also transferred to other modified concepts and instances by the application of the rules. If a concept A represents such a frame, the semantic priority is defined as a fuzzy membership function which depends on the number of realized parts in relation to the obligatory and optional parts for one modality description. This function takes into account that a hypothesis, i.e. an instance or a modified concept, does not become always more probable the more parts of a frame or a sentence are realized.

5.2 Statistical Modelling of Hypotheses Scores

The acoustic dissimilarity scores for word or word chain hypotheses provided by Dynamic Time Warping or Viterbi algorithms usually do not merely reflect the goodness-of-fit of the hypothesized unit but, moreover, the temporal duration of its acustical evidence. This can be seen by the fact that DTW distances or negative logs of HMM probabilities are in essence being built by summing up local, segmental distances, whatever subword unit these segments may represent (e.g. phonemes or centisecond labels).

A simple but useful model for this would be to neglect segmentation errors, or time registration effects resp. at all and to assume the **acoustic score** as the sum of independent random variables, identically distributed by (μ_t, σ_t^2) in case of correct hypotheses, or by (μ_f, σ_f^2) in case of failing hypotheses. If the duration of the acoustical evidence covered by the hypothesis is L (taken as the number of segments involved), the quality score q of correct hypotheses is distributed with

$$\mu_t(L)=E(q_t|L,\text{correct})=L\mu_t$$

and

$$\sigma_t^2(L)=\text{Var}(q_t|L,\text{correct})=L\sigma_t^2,$$

and the analoguous formula holds for the incorrect ones.

This linear dependence of quality score means and variances on the duration (if duration is measured in terms of the units used by the word matching process) as forecasted by our model has been empirically verified by statistical evaluation of word hypotheses scores generated by the EVAR word recognition module /SCH 87a/. In addition, the quality model means and variances have been estimated for this recognizer by a regression analysis.

Note that a scoring spread decreasing with L as stated above gives a good explanation to the frequent observation that reliability of word recognition seems to rise with their temporal duration. This fact is used to attach a **certainty score** to each hypothesis. The duration dependent Bhattacharyya distance between the quality distributions of correct and incorrect hypotheses reflects such a certainty. Under normal density assumption, the respective distributions are given with the above means and variances.

The judgement of a hypothesis is meant to help the overall system strategy to focus on promising partial solutions. Its value is usually computed on the basis of the quality score and it should reflect the hypothesis' chance of being part of the/a correct interpretation covering the whole utterance. In order to fullfill the admissibility requirements /NIL 82a/ of the A* graph search strategy that we use, judgement should always be an optimistic estimate of the optimal quality achievable by extending the partial interpretation in question.

Provided that the qualities are additive, this estimate is usually given as the sum of the hypothesis quality and an upper (lower in

case of dissimilarities) bound on the qualities of interpretations covering the remaining part of the utterance. If analysis is proceeding strictly from left to right, the shortfall method which estimates the remainder by the sum of segmentwise optimal qualities proves to be optimistic and therefore leads to an admissible search /WOO 82a/. If we make allowance for the by far larger search space using an island-driven strategy, admissibility can be achieved by the density method, i.e. by extrapolating the average within-hypothesis quality to the remaining utterance.

Our approach is to enforce a more goal-directed search as is possible with the "hypercautious" shortfall score by retaining the (relatively) small search space connected with the left-to-right conditions. For that, we include our knowledge concerning the statistical behaviour of the quality scores into the estimates mentioned above.

Let us assume the qualities of correct hypotheses to be distributed with $\mu(L)$, $\sigma^2(L)$, depending on their duration. Expectation and variance for the quality of an interpretation covering the by now uninterpreted part (of N segments duration) with correct hypotheses in general strongly depends on the assumptions of the statistical distribution of hypothesis duration. We are treating the three most simplifying cases:

Fine model: the remainder consists entirely of single-segment hypotheses, implying $\mu_{rem}(N)=N\mu(1)$ and $\sigma_{rem}^2(N)=N\sigma^2(1)$.

Coarse model: the remainder consists of exactly one N-segment hypothesis, implying $\mu_{rem}(N)=\mu(N)$ and $\sigma_{rem}^2(N)=\sigma^2(N)$.

Uniform block model: the remainder consists of N/B hypotheses of duration B each, implying $\mu_{rem}(N)=N/B\mu(B)$ and $\sigma_{rem}^2(N)=N/B\sigma^2(B)$.

Now we are able to loosen the requirement for an optimistic estimate in a reasonable fashion. As a **lower quality bound** for the not yet covered acoustic input we choose $s(N,c)=\mu_{rem}(N)-c\sigma_{rem}(N)$. The constant factor c is adjusted to satisfy a given probability of this bound being optimistic /SCH 87b, SAG 87a/.

Recalling our finding that the dependence of quality means and variances on hypothesis duration can be viewed as linear, the quality statistics of the remainder utterance reads as $\mu_{rem}(N)=\mu N$ and $\sigma_{rem}^2(N)=\sigma^2 N$, regardless of the duration model involved. With that, the quality bound reduces to $s(N,c)=\mu N-c\sigma N^{1/2}$, which guarantees a

rather deep search especially in the initial phase of analysis, when N is relatively large.

5.3 Outline of the Analysis Stratagy

As mentioned before, the analysis process starts with the selection of a few goal concepts out of the knowledge base. Beginning at the word level, modified concepts are constructed bottom-up by applying RULE4. This iterative process is based on the best few word hypotheses with respect to the acoustic score and the pragmatic priority. After the word hypotheses are chosen the process that modifies concepts up to the pragmatic level starts. It is guided by the concept priority vectors. This results in one successor node in the search graph for each selected word hypotheses and in further successors if a word hypothesis forms more than one potential goal concept.

The further expansion of nodes is done by applying the rules which build up the procedural semantics of the network language. If in one node more than one rule is applyable, the rules which create instances are prefered. The second selection criteria is the concept priority vectors of those concepts for which the rules can be applied. Note that there are instantiation processes which activate word hypothesing. The described recursive procedure leads to a search tree. The process is finished if a goal concept is instantiated and if the signal is mostly covered by the resulting interpretation.

The search strategy inside this tree is the A^*-Algorithm. Each node in an intermediate state of search is judged with respect to the actual modification of the goal concept which is associated with this node. In detail, for search tree nodes the following judgement vector is used

(structural consistency, acoustic score + quality bound,
certainty, pragmatic priority, semantic priority)

The first decision is whether in all the modified concepts and instances, which were created at the path to the actual node, satisfy the structural conditions. The second and third component are based on the statistical modelling of hypotheses scores. But in both cases not the exact value but only intervals are used. The comparison between two nodes is defined by the lexical order of their judgement

vectors. Exept the semantic priority all the components and therefore the lexical order of the vectors themselves fulfill the requirements of the A*-Algorithm for scoring functions.

6. CONCLUSION

After a detailled discussion of system architectures for knowledge based speech understanding, a homogenuous architecture based on semantic networks was develloped. The problem independent procedural semantics of this network language allows a flexible control of analysis processes. The A*-Algorithm builds the overall monitor of the system. Special judgement vectors reflect knowledge based and acoustic scores and they are admissible for the graph search procedure. So fare, the system was successfully tested in a prototype version. An efficient realization is presently under progress.

Acknowledgement: The work reported in this paper is supported by the German Research Foundation (DFG)

REFERENCES

/BAH 83a/ L.R. Bahl, F. Jelinek, R.L. Mercer: A Maximum Likelihood approach to Continuous Speech Recognition. IEEE Trans. PAMI 5 (1983) 179-190

/BRA 79a/ R.J. Brachman: On the Epistemological Status of Semantic Networks. In /FIN 79a/ 3-50

/BRA 85a/ R.J. Brachman, I.A. Schmolze: An Overview of the KL-ONE Knowldge Representation Language. Cognitive Science, Vol.9 (1985) 171-216

/BUC 85a/ B.G. Buchanan, E. Shortliffe: Rule Based Expert Systems. Addison-Wesley, Reading, Mass., 1985

/DEM 83a/ R. De Mori: Computer Models of Speech Using Fuzzy Algorithms. Plenum Press, New York, London, 1983

/EHR 87a/ U. Ehrlich: Multilevel Semantic Analysis in an Automatic Speech Understanding and Dialog System, Proc. Third Conference of the European Chapter of the Association for Computational Linguistics, Copenhagen, 1987

/ERM 80a/ L.D. Erman, V.R. Lesser: The Hearsay-II Speech Understanding System: A Tutorial. In /LEA 80a/ 361-381

/FIN 79a/ N.V. Findler (ed): Associative Networks. Academic Press, New York, 1979

/GOO 80a/ G. Goodman, R. Reddy: Alternative Control Structures for Speech Understanding Systems. In /LEA 80a/ 234-246

/HAY 82a/ F. Hayes-Roth, D.A. Waterman, D.B. Lenat (eds): Building Expert Systems. McGraw Hill, 1982

/HEW 71a/ C. Hewitt: PLANNER: A Language for Proving Theorems in

Robots. Proc. IJCAI 2 (1971)
/HEW 77a/ Viewing Control Structures as Patterns of Passing Messages. In: Artificial Intelligence 8 (1977) 323-364
/KIM 84a/ J.H. Kim, D.W. Payton, K.E. Olin: An Expert System for Object Recognition in Natural Scenes. Proc. IEEE 1st CAIA, Denver (1984) 170-175
/KOW 74a/ R. Kowalski: Predicate Logic as a Programming Language. Information Processing 74, North Holland, Amsterdam (1974) 569-574
/KUM 87a/ F. Kummert, H. Niemann, G. Sagerer, S. Schröder: Werkzeuge zur modellgesteuerten Bildanalyse und Wissensakqusition - Das System ERNEST. In M. Paul (Hrsg.): GI - 17. Jahrestagung Computerintegrierter Arbeitsplatz im Büro. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1987, 556-570
/LEA 80a/ W.A. Lea (ed): Trends in Speech Recognition. Prentice-Hall, Englewood Cliffs, N.J., 1980
/LEV 79a/ H. Levesque, J. Mylopoulos: A Procedural Semantics for Semantic Networks. In /FIN 79a/ 93-121
/LOW 80a/ B. Lowerre, R. Reddy: The Harpy Speech Understanding System. In /LEA 80a/ 340-360
/MAN 74a/ Z. Manna: Mathematical Theory of Computation. McGraw-Hill, New York, 1974
/MCC 69a/ J. McCarthy, P.J. Hayes: Some Philosophical Problems from the Standpoint of Artificial Intelligence. In B. Melzer, D. Ritchie (eds): Machine Intelligence 4. Edinburgh Univ. Press, Edinburgh, 1969, 463-502
/MCC 81a/ D.L. McCracken: A Production System Version of the Hearsay-II Speech Understanding System. UMI Research Press, Ann Arbor, Michigan, 1981
/MCC 83a/ G. McCalla, N. Cercone (eds): Knowledge Representation. IEEE Computer Magazine (1983)
/MER 85a/ G. Mercier: Expert Systems Approach to Acoustic-Phonetic Decoding and Word Recognition. In H. Niemann (ed): Mustererkennung 85. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985, 290-312
/MIN 75a/ M. Minsky: A Framework for Representing Knowledge. In P.H. Winston (ed): The Psychology of Computer Vision. Mcgraw-Hill, New York, 1975, 211-277
/MYL 83a/ J. Mylopoulos, H. Levesque: An Overview of Knowledge Representation. In M. Brodie, J. Mylopoulos, J.V. Schmidt (eds): On Coceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1983
/NIE 81a/ H. Niemann: Pattern Analysis. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1981
/NIE 85a/ H. Niemann, A. Brietzmann, R. Mühlfeld, P.Regel, G. Schukat: Th Speech Understanding and Dialog System EVAR. In R. De Mori, S.Y. Sun (eds): New Systems and Architectures for Automatic Speech Recognition and Synthsis. NATO Series F 16. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985, 271-302
/NIE 86a/ G. Niedermair: Divided and Valency-Oriented Parsing in Speech Understanding. Proc of the 11th International conference on Computational Linguistics, Bonn (1986) 593-595
/NIL 82a/ N.J. Nilsson: Principles of Artificial Intelligence. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1982
/PRA 85a/ H. Prade: A Computational Approach to Approximate and Plausilble Resoning with Applications to Expert Sytems. IEEE Transactions on Pattern Analysisand Machine Intelligence PAMI-7

(1985) 260-283
/RAB 88a/ L.R. Rabiner: Mathematical Foundations and Applications of HMM. This volume
/SAG 85a/ G. Sagerer: Darstellung und Nutzung von Expertenwissen für ein Bildanalysesystem. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985
/SAG 87a/ G. Sagerer, F. Kummert, E.G. Schukat-Talamazzini: Flexible Steuerung eines sprachverstehenden Systems mit Hilfe mehrkomponentiger Bewertungen. In E. Paulus (Hrsg.) Mustererkennung 1987. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1987, 123-127
/SCH 86a/ P. Schefe: Künstliche Intelligenz - Überblick und Grundlagen. Bibliographisches Institut, Mannheim, Wien, Zürich, 1986
/SCH 87a/ E.G. Schukat-Talamazzini: Generierung von Worthypothesen in kontinuierlicher Sprache. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1987
/SCH 87b/ E.G. Schukat-Talamazzini: Private Communication
/SCH 88a/ R.M. Schwartz: Acoustic Phonetic Decoding of Speech. This volume
/THU 88a/ G. Thurmair: Semantic Analysis of Continuous Speech. This volume
/TSO 80a/ J.K. Tsotsos, J. Mylopoulos, H.D. Covvey, S.W. Zucker: A Framework for Visual Motion Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence Pami-2 (1980) 563-573
/WIN 83a/ T. Winograd: Language as a Cognitive Process. Volume 1: Syntax. Addison-Wesley, Reading, Mass., 1983
/WOL 80a/ J.J. Wolf, W.A. Woods: The HWIM Speech Understanding System. In /LEA 80a/ 316-339
/WOO 70a/ W.A. Woods: Transition Network Grammars for Natural Language Analysis. Comm. ACM 13 (1970) 591-606
/WOO 75a/ W.A. Woods: What's in a Link? Foundations for Semantic Networks. In D.G. Bobrow, A. Collins (eds): Representation and Understanding. Academic Press, New York, 1975
/WOO 82a/ W.A. Woods: Optimal Search Strategies for Speech Understanding Control, Artificial Intelligence 18 (1982) 295-326
/WAT 86a/ D.A. Waterman: A Guide to Expert Systems. Addison-Wesley, Reading, Mass., 1986