언 어

Korean Journal of Linguistics

제 21 권 제 1,2 호 (Vol. 21, No. 1,2)　1996년 6월 (June 1996)

한 국 언 어 학 회

The Linguistic Society of Korea

A Database Interpretation of Natural Language

Roland Hausser

(Friedrich-Alexander-Universität Erlangen-Nürnberg)

Hausser, Roland. 1996. **A Database Interpretation of Natural Language.** *Korean Journal of Linguistics*, 21-12, 29-55. The goal of the LAG[1] theory of grammar and the SLIM[2] theory of language is to model the information transfer in communication as an objectively functioning system such as a talking robot. The SLIM theory models the mechanics of natural language communciation as a Surface compositional, time Linear, Internal Matching procedure. This paper describes, how the SLIM theory of language is formally realized as an extension of a classic database.

Section 1 investigates the relevant differences and similarities between (i) the interaction of a user with a database and (ii) the interaction between two speaker/hearers. Section 2 compares six basic methods of accessing the context of use to find a suitable primary key for storing and retrieving information. Section 3 analyzes the context formally as a network database, called LA-base, into which propositions are embedded in the novel format of 'distributed, bidirectional, co-indexed tokens.' Section 4 reconstructs the intuitions and concepts of the SLIM theory of language within the structure of LA-base. Section 5 describes the pragmatic interpretation in LA-base, both in the hearer- and the speaker-mode, and shows how LA-base may function as the relevant component of a robot communicating in natural language. **(Friedrich-Alexander-Universität Erlangen-Nürnberg)**

1. Databases and context of use

In order to function objectively, a computerlinguistic model of natural language communication requires an explicit definition of the context of use. What, however, are the exact ontological, structural and computational properties of this component? As a first, most general

---

[1]LAG is an acronym for 'Left-Associative Grammar,' designed to model the basic time-linear structure of natural language. The formal definition and complexity analysis of LAG is published in Hausser 1992. For extensive applications to natural language see Hausser 1986 and 1989.

[2]A preliminary description of this theory is published in Hausser 1989:27-30, 107-118, 267-287, 295-317. This paper is based on chapter 23 of a systematic description of SLIM (unpublished).

answer to this question we have called the context of use a speaker/hearer-internal *database*[3] in the widest sense of the word.

This provides not only a computational framework for the representation of individual knowledge, but also a functional analogy for natural language communication. For the operations of *storage* and *retrieval* in databases resemble the cognitive procedures in the hearer and the speaker in important respects.

An earlier attempt[4] to realize this analogy in terms of a program using frames resulted in a number of difficulties, however. Especially the time-linear embedding and extraction of language meaning into and out of the context raised difficulties for the formal implementation.

Conceptually, these problems are caused by treating examples of language meaning and context as closed units, positioned accross from each other to illustrate matching and extraction within the SLIM theory of language. This holistic way of representing the two levels is suited to explain the elementary mechanics of internal matching pragmatics.[5] But it does not provide a sufficiently abstract format which would support the intended time-linear correlation of language meaning and context of use in a structurally simple manner.

In order to develop such a format from the viewpoint of databases, let us compare the use of a conventional database (DB interaction) with corresponding aspects of natural language communication (NL communication).

1.1. DB interaction and NL communication

- ENTITIES INVOLVED

  Database interaction

  The interaction takes place between two completely different entities, namely the user and the database.

  NL communication

  The interaction takes place between two similar and equal cognitive agents, the speaker and the hearer.

---

[3]Hausser 1989:29f.

[4]Hausser 1989, chapters 4 and 5.

[5]Hausser 1978, 1979a,b, 1981, 1983, 1984, 1987.

• ORIGIN OF CONTROL

Database interaction

The control of database operations such as input and output is completely in the hands of the user.

NL communication

There is no user. Instead, cognitive agents control each other in their joint communication by alternating in the speaker- and hearer-mode (*turn taking*).

• METHOD OF CONTROL

Database interaction

The user controls the operations of the database with a programming language, the commands of which are executed as electronic procedures.

NL communication

The speaker controls the *generation* of language as an autonomous agent, coding the parameters of the utterance situation into the output expressions. The hearer's *interpretation* is controlled by the incoming natural language expression.

• TEMPORAL ORDER

Database interaction

The output (database as 'speaker') occurs neccessarily **after** the input (database as 'hearer').

NL communication

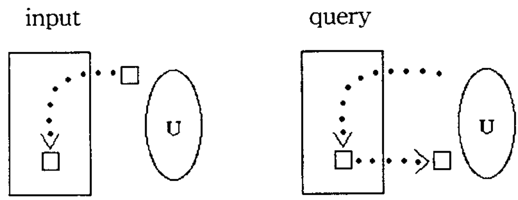
Verbalizing (output procedure of the speaker) occurs necessarily **before** the interpretation (input procedure of the hearer).

The basic differences between database interaction and NL communication are summarized schematically in 1.2.:[6]

---

[6]The small boxes in 1.2. represent the language sign serving as input and output. The 'u' indicates the user.

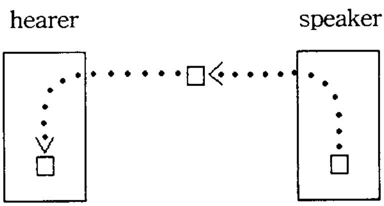
1.2. Interaction with a conventional database



moment of time t_1 $<$ t_2

database A $=$ A

1.3. Interaction between speaker and hearer



moment of time t_2 $>$ t_1

database H $\neq$ S

From the fact that NL communication practically always uses two different databases for verbalizing and interpretation,[7] namely the context of the speaker and the hearer, respectively, follows the characteristic notion of successful communication. A natural language communication between a speaker and hearer is successful only if the expressions refer to *corresponding* substructures of the respective databases.[8]

---

[7]Except when someone talks to oneself. This case should be analyzed as a verbalizing without a hearer, rather than a language controlled access to certain substructures of ones own context.

[8]For example, the postcard example in Hausser 1989:274 does not refer to an arbitrary dog or an arbitrary kitchen. Instead, the communication between the author and the addressee of the postcard can be regarded as successful only if both refer to *corresponding* subcontexts which represent the *same* referents.

2. Possible structurings of the context

Controlling the access to the correct substructure of the speaker/hearer's context raises the question of the PRIMARY KEY,[9] i.e. the structural principle according to which informations are stored and retrieved. In the natural languages, the following structural parameters contribute to the control of storage and retrieval:

2.1. Components of the key

- Index
  On the level of semantics, the different words of a proposition are held together by their common index (proposition number). Pragmatically, the index serves as STAR-point.[10]
- Grammar
  The words and expressions of a proposition form a grammatical structure. Thereby the different roles of the parts (valency carrier, valency filler of a certain case, modifier) contribute to the control of possible continuations within a proposition at a given STAR- point.
- Word
  The choice of content words in a proposition determines which topics within a context structure are accessed.

Because each of these three components has a specific function, we would like to formally combine them in a way that allows optimal storage and retrieval in database-like context structures of natural language.

The simplest and most transparent method of such a combination is to execute the three structural aspects I (index), G (grammar) and W (word) as separate steps one after the other. This method results in six possible orders:

---

[9]Cf. Date 1990:87 ff.

[10]The STAR-point stands for the parameter values of space, time, agent and recipient, relative to which natural language expressions are interpreted. See Hausser 1989:272ff.

2.2. Six possible methods of structuring context access

I -- G -- W	G -- W -- I	W -- G -- I
I -- W -- G	G -- I -- W	W -- I -- G

Each method of access presupposes a corresponding structure of the database being accessed. In this sense, the six possible methods of structuring access in 2.2. also represent six elementary ways of structuring the context of use.

At first glance, a meaningful choice between the alternatives in 2.2. may seem difficult. However, the choice is facilitated by the fact that some of the alternatives in in 2.2. have already been used in the literature -- though for other apparent purposes than accessing the context of use.

The choice of the index 'I' as the primary key is in keeping with the physicalist view of analytic philosophy and model-theoretic semantics, which classifies states of affairs according to their occurrence in space and time. An 'I--G--W' approach may be found, for example, in the logical model theory of Montague. Within this framework, a sentence like *Fido likes Zach* is represented as the logical proposition

```
like'(fido',zach')
```

This expression does not have a truth value *per se*, but only relative to a presupposed model structure.[11] In Montague's PTQ this relativization is expressed by means of a superscript (here '$^{@,i_1,j_1}$'):

2.3. Example of an I--G--W analysis

$$\texttt{like'(fido',zach')}^{@,i_1,j_1}$$

The @ in 2.3. names the model structure and i_1,j_1 is the space-time index specifying the specific model intended for the interpretation. After

[11]See Hausser 1989:28, 2.3.2., for an explicit example.

selection of the index ('I'), the grammatical structure ('G') of the logical formula is interpreted step by step:

> **like'(fido',zach')**$^{@,i_1,j_1}$ ist 1 if and only if **(fido',zach')**$^{@,i_1,j_1}$ is an element of **F(like')**(i_1,j_1).

This process of decomposing is continued until only elementary constants ('W') remain, such as **like'**, **Fido'** and **Zach'**. Once their denotations have been determined in the model structure at the chosen index (i_1,j_1) via the denotation function F, the rules of semantic interpretation are applied in the inverse direction to determine the truth value of the complex proposition.

In logical model theory the index serves as the primary key insofar as it establishes the relation between the logical proposition and the intended model. The goal of logical model theory is to derive the truth value of arbitrary propositions relative to *arbitrary indices* in arbitrary model structures. A realistic model of natural language communication, on the other hand, requires an index structure which helps to infer the *correct* speaker/hearer-internal subcontext for each utterance.

A second possible choice of a primary key is the grammatical structure 'G'. This choice corresponds to the viewpoint of generative grammar, which aims at characterizing natural language syntax by establishing a general typology of sentence frames. A 'G--W--I' approach was explored in the attempt to define a suitable context of use for the pragmatic interpretation of the LA-grammatical fragment of English in Hausser 1989.[12]

According to the 'G--W--I' method, the primary key 'G' divides the context into subsections for different sentence frames, in which the instantiations of the respective sentence frames are stored. Within a subsection, sentences of equal structure are distinguished in terms of their differing choice of words ('W'). Finally, different readings of syntactically ambiguous sentences are distinguished by different indices ('I').

---

[12]That the frame-theoretic formalism used for the semantic representation of sentences should be used also for the representation of the context of use follows from the goal of embedding/extracting meanings as directly as possible within the SLIM theory of language.

Applications using the primary keys 'I' and 'G' in general, and the 'I--G--W' and 'G--W--I' methods in particular are suboptimal choices for the following reasons. Regarding the interpretation of natural language, they pose the problem that their respective index cannot function as a STAR-point and thus cannot contribute to finding the correct subcontext during pragmatic interpretation. Regarding the generation of natural language, they pose the problem that their respective contexts do not provide structural support for a time-linear navigation.

The third and last possibility is using the words 'W' as the primary key. It is this approach which we will use for the semantico-pragmatic interpretation of LA-grammar. Preliminary investigations of the six alternatives stated in 2.2. have shown that the 'W--I--G' method is suited best to simultaneously satisfy the numerous empirical principles of natural language semantics and pragmatics within the SLIM theory.

A context of use which is structured according to 'W--I--G' method is called an 'LA-base.' This name is intended to show that such a context is a database-like structure which uses a left-associative grammar (LA-grammar) to interpret and generate expressions of natural language in a time-linear fashion.

Pretheoretically, LA-base has the form of a table of *lemmata*.

2.4. W--I--G analysis as LA-base

word x: i_1 [], i_5 [], i_9[] ... i_27 [functor: word_y]

word y: i_3 [], i_7 [], ... i_27 [argument: word_x]

Each lemma consists of the base form of a content word (in 2.4. for example word_x) and a sequence of 'tokens' i_n[ ], whereby i_n is an index and [ ] represents a feature-value structure. The base forms of the lemmata serve as the primary key 'W' and are ordered alphabetically.

Each token of a lemma has a unique index, which serves as the secondary key 'I' of the 'W--I--G' method. The index names the proposition of which the token is a part. The propositions in LA-base consist of tokens which occur in various different lemmata, but share the same index.

The feature-value structure of a token serves as the tertiary key 'G.' It describes (a) the grammatical properties of the token, for example NUMERUS: SINGULAR, and (b) names the base forms of those words which serve as higher functor (e.g. VERB: *like*) or as lower argument (e.g. NOMINATIVE: *Fido*) in the proposition associated with the index of the token.

The index i_27 in 2.4. connects two tokens in two different lemmata as parts of the same proposition. The token i_27[ ] of word_x names word_y as a higher functor and the token i_27[ ] of word_y names word_x as the associated argument in the proposition with the index i_27. This is the format of **distributed bidirectional co-indexed tokens** characteristic of LA-base.

At first glance, LA-base seems to be a simple lexicon structure. Yet its format is suited to code the syntactico-semantic structure of any type of proposition (e.g. episodic or absolute) in terms of co-indexed tokens and their feature-value structure. Furthermore, this format provides the structural basis for the most elementary form of time-linear navigation through the context: any token at any position in LA-base explicitly specifies the intra-propositional continuations in terms of the feature-value structure and the index.

The automatic construction of (the feature-value structures of) the tokens is performed by the syntactic rules of LA-grammar. The 'W--I--G' representation of propositions in the form of co-indexed distributed tokens replaces the earlier 'G--W--I' interpretation of LA-grammar (Hausser 1989) in the form of more traditional tree structures (illustrated on the left hand side of 2.5.).

The 'G--W--I' and the 'W--I--G' approach in LA-grammar have in common that they are based on a strictly time-linear derivation order. For this reason there is a systematic transition from a 'G--W--I' to a 'W--I--G' interpretation:

2.5. Transition from G--W--I to W--I--G

	sentence$_{23}$		$\longrightarrow$	Fido (name): ...i_{23} Function: subject Higher functor: like
Subject	verb	object		
name	**like$_{23}$**	name		like (verb): ...i_{23} Function: sentence Arguments: Fido, Zach
Fido$_{23}$		**Zach$_{23}$**		Zach (name): ...i_{23} Function: Object Higher functor: like

The distributed, bidirectional format of the 'W--I--G' analysis on the right of 2.6. expresses all the grammatical properties expressed in the 'G--W--I' analysis on the left. The 'W--I--G' analysis, however, has the additional advantage of providing the structural foundation for a time-linear navigation.[13] It is often possible, however, to transform the distributed grammatical structure of a 'W--I--G' analysis into the holistic structure of a corresponding 'G--W--I' tree.

3. LA-base as a network database

The 'W--I--G' format of LA-base may be interpreted as a 'classic' database with 'records.'[14] A *record* is a data type constructed in accordance with a strictly defined pattern. This pattern is called the *record type*, consisting of a name and a number of attributes. Several different record types may be used in a database.

The tokens of LA-base are defined as records, whose types specify attributes for the index and a number of grammatical features. For the four different kinds of content words, LA-base specifies the following record types:

[13]In SLIM, generation of natural language is based on a time-linear path through the database. See Hausser 1989:108.

[14]This is in contrast to the frame-based context structures considered in Hausser 1989, which belong to the object-oriented type of 'non-classic' databases. See Elmasri & Navathe 1989 as well as Date 1990[v].

3.1. Different *record types* of tokens in LA-base

```
T_Noun
Index  Quantifier  Case  ADJECTIVE  VERB

T_Adjective
Index  Comparation  NOUN

T_Intransitive_Verb
Index  Modus  NOMINATIVE

T_Transitive_Verb
Index  Modus  NOMINATIVE  ACCUSATIVE
```

The first *record type* in 3.1., for example, has the name 'T_Noun' and the attributes 'Index Quantifier Case ADJECTIVE VERB.'

A database definition must specify the 'data type' for each attribute of each record type. The data type of an attribute determines what kind of value it can have and how much space it may take. Regarding the data types of LA-base, we are content for the moment with a very general specification in terms of the distinction between LOGICAL ATTRIBUTES and CONTINUATION ATTRIBUTES.

Logical attributes are identified by lower case letters and take abstract values, like numbers for indices or fixed letter sequences for categorial properties. Continuation attributes, on the other hand, are identified by upper case letters and take the base forms of content words as their value. Together with the index of the token, a base form value specifies a possible continuation within the proposition at hand.

A record results by assigning values to the attributes of a record type, as in 3.1., which illustrates a record of the type T_Noun:

3.2. Example of a T-record in LA-base

index	quantifier	case	ADJEKTIV	VERB
i23	def, sg	N	good	read

The records of a given record type are collected in a table, called a 'record structure.' Listing thousands of records in the associated record structure provides a simple structural basis for the usual operations of storage, retrieval and update in classical databases.

The lemmata of LA-base code not only the *tokens* as records, but also the *base forms*. For the base forms of the four different kinds of content words, LA-base specifies the following record types:

3.3. Different *record types* of base forms

```
G_Noun
base_form  part_of_speech  gender  declination_class

G_Adjective
base_form  part_of_speech  comparation_class

G_Intransitive_Verb
base_form  part_of_speech  conjugation_class  valency_structure

G_Transitive_Verb
base_form  part_of_speech  conjugation_class  valency_structure
```

The attributes of the base form records (hence G-records) differ from those of corresponding token records (hence T-records). G-records specify the base form where T-records specify the index. Also, there are no continuation attributes in G-records.

In the literature, classic, record-based databases are divided into three basic types, namely relational, hierarchical and network databases. Of these, LA-base is of the type network database. A network database defines a 1:n relation between two record types, called the *owner records* and the *member records*.[15]

For example, the different departments of a university may be defined as owner records and the students majoring there as member records:

---

[15]The following description of network databases follows the conventions of DBTG (Database Task Group) in general and Elmasri & Navathe 1989 in particular.

3.4. Example of a network database

owner record	*member records*			
Comp.Sci.	Riedle	Schmidt	Stoll	...
Mathematics	Müller	Barth	Jacobs	...
Physics	Weber	Meier	Miele	...

In this simplified example, the different records are represented by names. In an explicit database, the owner record type 'department' would specify attributes like name, address, phone number, etc. while the member records type 'student' would specify attributes to characterize each person.

The number of member records for a given owner record is variable in a network database. A department in 3.4. could have no students at all -- when it has just been founded, for example. Maintaining the 1:n relation between an owner and its member records requires, however, that any given member record is assigned to only one owner. This means that in example 3.4. no student may have more than one major.

The 1:n relation between two record types in a network database is called a set type. Instantiations of a set type are called set instances. For example, 3.4. illustrates three *set instances* of the same *set type*.

In LA-base, each lemma constitutes a set instance, whereby the G-record of the base form is the owner and the T-records of the associated tokens are the members. In LA-base there are the following examples of different *set types*:

3.5. *Set types* in LA-base

owner types	*member types*			
G_Noun	T_Noun	T_Noun	T_Noun	...
G_Adjective	T_Adjective	T_Adjective	T_Adjective	...
G_IT_Verb	T_IT_Verb	T_IT_Verb	T_IT_Verb	...
G_Tr_Verb	T_Tr_Verb	T_Tr_Verb	T_Tr_Verb	...

In LA-base, the member records assigned to a given owner all have the same type (i.e. the tokens in a given lemma all have the same attribute structure). For this reason LA-base is an especially simple kind of network database with so-called *single member sets*. Furthermore, because the owner types are in principle defined to be distinct from the associated member types (cf. 3.1. and 3.3.), LA-base is a *non-recursive* network database.

The fixed structure of the owner records in an LA-base is derived directly from the lexicon of the language to be handled: after selecting the categorized content words from the lexicon, the G-records are automatically generated from the categories. The lemmata (*set instances*) in LA-base are listed in the alphabetical order of their base forms.

The variable structure of the member records, on the other hand, is created automatically by the rules of LA-grammar while they apply in the process of parsing natural language expressions, e.g., a text. For example, given the sentence *Fido likes Zach*, the rules of the LA-grammar for English derive a T-record for each of the three content words. These three T-records share the same index, but differ in their record types and their values. Adding new T-records in an LA-base is simple, because they are always added at the end of the respective lemmata.

3.6. Example of a proposition coded in LA-base

```
owner (G-records}          members (T-records)

[base_form:       Fido       ...          [Index: i
part_of_speech: name]                      Case:  Nominative
                                                  ADJECTIVE: #
                                                  VERB: like]
[base_form:       like       ...               [Index: i
part_of_speech:  Verb                            Modus: Indicative
Valenzstr: Nom,  Acc]                            NOMINATIVE: Fido
                                                 ACCUSATIVE: Zach]
[base_form:       Zach       ...    [Index: i
 part_of_speech: name                Case: Accusative
                                     ADJECTIVE: #
                                     VERB: like]
```

The co-indexed tokens in 3.6. describe a proposition, whereby the semantic functor/argument structure is expressed in terms of the feature/value structure of the tokens involved. The partial LA-base 3.6. corresponds to the right hand side of 2.6., whereby the feature-value structures of 2.6. have been replaced by the more detailed specification of owner and member records in accordance with the principles of network databases.

4. Linguistic interpretation of LA-base

The basic structure of LA-base complies with the known formalism of a non-recursive network database with single member sets. This fact is helpful in the theoretical analysis and computational implementation of LA-base, because storage, retrieval and update operations known from conventional network databases can be reused.

At the same time, LA-base goes beyond a conventional network database because it explicitly defines possible continuations.[16] These provide the basis for an operation not known in conventional databases, namely a linear navigation through the database which runs orthogonal to the basic access structure of owner and member records.

The structure of possible continuations in LA-basis is completely general: the distributed, bidirectional format of co-indexed tokens allows the representation of any type of proposition. This generality of the formal system raises the question of how the specific concepts and procedures of the SLIM theory of language should be realized within LA-base. The answer may be devided into the follow three parts:

4.1. Linguistic tasks of LA-base

1. The distinction between (i) the semantic representation of language expressions and (ii) the context of use -- crucial for the functioning of natural language communication -- must be realized in LA-basis.

[16]Illustrated in 3.6. with the example of an intrapropositional continuation.

2. The semantic and pragmatic generation and interpretation must be defined in terms of explicit algorithms, which permit the correct handling of natural language expressions in LA-base automatically, i.e. without any need for additional control in terms of external programmed commands.
3. It must be shown how LA-base can function as the language and database component of a robot.[17] Of special interest for a general theory of language is thereby the interaction between elementary concepts defined as classes of perception parameter values[18] and the characteristic compositional semantics of LA-base.

The task mentioned in 4.1. are closely interrelated, both in terms of their structure and their function. Within the formal framework of LA-base, they are realized as follows.

The basic elements of LA-base are the content words (owner records) and their tokens (member records). The first attribute of each token record is the index. The indices of LA-base -- in their most elementary function -- are simply numbers which specify whether or not two tokens belong to the same proposition in terms of the (in)equality of their respective indices. From a linguistic point of view, however, there are important relations *between* propositions. To formally express the content of such inter-propositional relation we superimpose a differentiated *index structure* on top of the basic index numbers.[19]

The first step in a linguistic interpretation of the indices of LA-base consists in defining different classes of propositions in terms of different IndexTYPES. The following classes of propositions are necessary for the pragmatic interpretation and the associated inferencing in LA-base.

4.2. Three classes of propositions in LA-base

1. NON-VERBAL PROPOSITIONS of the context
   (a) ABSOLUTE PROPOSITIONS

[17] Hausser 1989:295-331.
[18] cf. Hausser 1989:316.
[19] Methodologically, this step resembles the definition of possible continuations within the conventional feature/value structure of token records in section 3.

Propositions in this class are statements which describe the meaning of content words, e.g. *A square has four equal sides* or *A table is a piece of furniture*. Asolute propositions are identifiable by their token indices of the form A_n, whereby n is an arbitrary number.

(b) EPISODIC PROPOSITIONS

Proposition of this class are statements which describe content words in episodic connections, e.g. *Julia bought a table.* Episodic propositions are identifiable by their token indices of the form E_n.

2. VERBAL PROPOSITIONS

Propositions of this class characterize the literal meaning (semantic representation) of natural language expressions, e.g. *I bought a table yesterday.* Verbal propositions are identifiable by their token indices of the form V_n.

Within the database structure of LA-base, these three classes of propositions are distinguished by storing their respective tokens in three different areas of the member records of their lemmata, in accordance with the index type.

4.3. Index structure of a lemma in LA-base

	member records:		
	Typ 1a: A_n	Typ 1b: E_n	Typ 2: V_n
owner record:			
base form	absolute propositions	episodic propositions	verbal propositions

Apart from their different places of storage, the three types of propositions have the same general structure. Differences show up only in the *values* of the tokens.

The index values A_n, E_n and V_n have different interpretations in LA-base. The content of absolute propositions is regarded as valid independently of time and space, for which reason the indices A_n are not restricted in their space-time interpretation. The content of episodic propositions refers to specific events, for which reason the indices E_n must be related to specific space-time intervals by means of an interpretation table. In verbal propositions, finally, the index V_n must be related to the correct parameters of space, time, modality etc. by interpreting adverbials, the tense and modality of the verb, personal pronouns, etc.

Which index type is assigned to the analysis of an input sentence depends on its structure and its content. Non-verbal propositions differ from verbal propositions in that non-verbal propositions are in principle *semantically complete*. This means that all features in the tokens of a non-verbal proposition have direct values for time, place, modality, verbal arguments, etc. In this sense, non-verbal propositions correspond to the notion of 'logical' propositions.
Verbal propositions, on the other hand, may contain pronouns, ellipses, tense and modal operators, and adverbials, for which reason not all of the intended arguments and relations can be entered into the feature-value structure of the tokens as direct values. This 'semantic incompleteness' necessitates the pragmatic interpretation of verbal propositions, which has the task of turning them into corresponding, semantically complete, non-verbal propositions. Thereby the original index V_n must be mapped into a correct non-verbal index.

For the SLIM theory of language, non-verbal propositions constitute the context of use. 4.4. shows how the levels of (1) context and (2) literal meaning are related to each other in LA-base.

4.4. Meaning and context in LA-base:

base form of content words *(owner records)*

tokens: 1. non-verbal propositions ⟷ 2. verbal propositions *(member records)*

1a. absolute propositions 1b. episodic propositions **internal matching pragmatics** in LA-base

In the hearer mode, the rules of pragmatics embed verbal propositions into this context (non-verbal propositions) of use by making them semantically complete. In the speaker mode, the rules of pragmatic interpretation extract verbal propositions from the context of use by coding their STAR-point into the surface by means of pronouns, tense and modal operators, etc.

4.3. showed how the two levels of (1) the context of use and (2) the literal meaning of language expressions are formally distinguished in LA-base. 4.3., on the other hand, shows how these two levels are related to each other -- in concord with the basic function of the SLIM theory of language. This completes our answer to question 1 of 4.1.

5. Proposition and concept in LA-base

Next we turn to the second area of 4.1., namely the question of how the semantic and pragmatic interpretation of natural language expressions in LA-base should be defined as a concrete algorithm. The basis for this definition is the (2+1) level structure of the SLIM heory of language, namely

(i) the SURFACE of the expressions,
(ii) their VERBAL PROPOSITION (semantic representation) and
(iii) the corresponding NON-VERBAL PROPOSITION (context).

Between these three levels of representation there function two bidirectional algorithms in vertical direction, namely

(V1) the SEMANTIC INTERPRETATION and
(V2) the PRAGMATIC (DE-)VERBALIZATION.

In addition there are two unidirectional algorithms, which function at the levels (i) und (iii) in the horizontal, time-linear direction, namely

(H1) the SYNTACTIC ANALYSIS of the surface and
(H2) the PRAGMATIC INFERENCES on the basis of non-verbal propositions.

The correlation between the three levels as well as their two vertical and their two horizontal algorithms are represented schematically in 5.1.:

5.1. Levels and algorithms of interpretation

(i) surface
  → syntactic analysis (H1)
↕ semantic interpretation (V1)
(ii) verbal proposition (semantic representation)
↕ pragmatic (de-)verbalizing (V2)
  → pragmatic inference (H2)
(iii) non-verbal proposition (context)

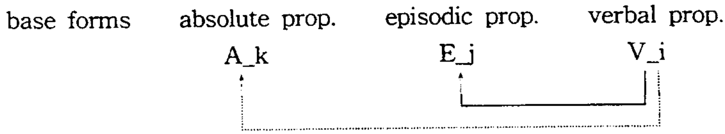
The interpretation of natural language is driven by the unidirectional algorithms H1 und H2. In the hearer-mode, it is the time-linear syntactic analysis (H1) of the incoming stream of language. In the speaker-mode, it is the time-linear pragmatic inferencing (H2) at the level of non-verbal propositions.

The motor algorithms H1 and H2 may run without being connected to the process of language interpretation. In case of syntactic analysis H1, this corresponds to a hearer who analyses the incoming surfaces grammatically without understanding them, i.e. without deriving the associated verbal and non-verbal propositions. In case of pragmatic inference H2, on the other hand, this corresponds to a speaker, who follows his thoughts, i.e. navigates through the non-verbal context, without verbalizing them.

If LA-base is used for the interpretation of natural language input, however, then the semantic clauses (V1) in the syntactic rules (H1) of the LA-grammar derive verbal propositions (semantic representations). These are usually semantically incomplete and must be mapped by the rules (V2) of pragmatic (de-)verbalizing into corresponding non-verbal propositions, whereby the original V_n index is mapped into a corresponding E_n index. The procedure of pragmatically deverbalizing a verbal proposition automatically defines a time-linear path through the context of use.

In its most basic form, the pragmatic interpretation in the hearer-mode may be regarded as a REORDERING of verbal propositions in the course of their semantic completion into non-verbal propositions.

5.2. Schema of pragmatic interpretation in the hearer

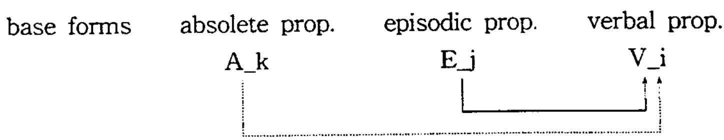


The region of verbal propositions serves as a 'launch pad.' It is cleared after mapping a given input into a non-verbal proposition, to be ready for the next input or -- after turn taking -- output.

The generation of language in LA-base originates with a navigation through the non-verbal context H2 (following a train of thought). This path through the context automatically defines non-verbal propositions. By applying the rules of pragmatics V2 in the inverse direction ↑, these non-verbal propositions are mapped into verbal propositions which are realized as language surfaces.

In its most basic form, the pragmatic interpretation in the speaker-mode may be regarded as a REORDERING of non-verbal propositions in the course of their semantic de-completion into verbal propositions.

5.3. Schema of pragmatic interpretation in the speaker



The region of verbal propositions serves as launch pad also in the speaker mode. It is cleared after mapping a non-verbal proposition into a verbal one and realizing it as a surface.

Verbal propositions are derived from semantically complete non-verbal ones, because only complete propositions allow the speaker to navigate freely through the context, i.e. independently of any properties of sentences that were interpreted before. Semantically complete, non-verbal propositions are made semantically incomplete by the speaker, because the use of pronouns, tense and modal operators, etc. in the natural language surface allows the hearer to deduce the intended relation between the propositional content and the utterance situation (STAR-point), which is a precondition for storing the content as the non-verbal proposition.

This brief outline of the pragmatic interpretation in LA-base in the hearer- and the speaker-mode provides the anwer to question 2 of 4.1.: the pragmatic interpretation in LA-base is a mapping with a clearly defined domain and range. The definition of this mapping is structurally clear and models the mechanics of natural language communication in a conceptually meaningful, functional way.

Finally we turn to question 3 of 4.1. -- the possible use of LA-base in a robot communicating in natural language. Because LA-base was developed from a classic database structure, the interaction between the user and LA-base resembles in many respects the interaction with a conventional computer. This means that input and output is possible only at the level of language, because the system has no non-verbal interaction with its task environment -- in contrast to a robot of the 3. generation.[20]

As a piece of software, LA-base is special, however, in that it uses a natural language for input and output rather than the special commands of a programming language. In addition, this software is designed

1. to analyze natural language automatically as database information and to correctly store it;

2. to navigate a focus point through the database based on a time-linear reasoning by the autonomous cognitive agent;

3. to map the information traversed during navigation into natural language expressions.

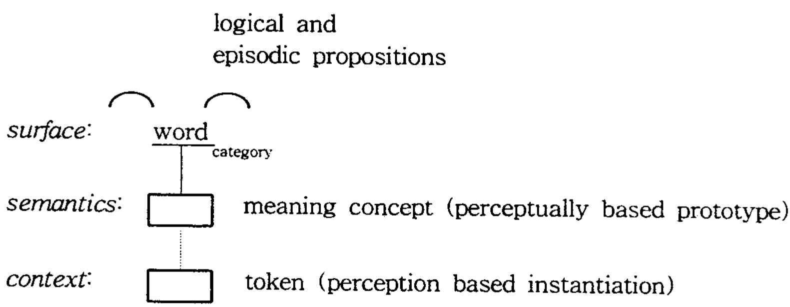
Before this software can be used as the verbal and non-verbal database

[20]See for example Wloka 1992.

component of a robot, however, it must have the additional ability to access and to use the perceptual data of the robot.

This extension concerns the theoretical relation between the propositional meaning aspect of LA-base and the conceptual meaning aspect of the perceptual component of the robot. Within the SLIM theory of language, this relation may be characterized as follows:

5.4. Propositional and conceptual aspects of meaning



5.4. indicates the propositional role of a content word by means of the two arcs to the left and right of the surface, which are intended to represent the compositional interaction of the word with other words in the proposition. The conceptually based meaning aspect of a content word, on the other hand, are located within the boundaries of the word and are represented by the box underneath of and attached to the surface.

In other words, the propositional and the conceptual aspect of a content word are treated separately from each other within the SLIM theory of language.[21] The literal meaning of a content word ('word semantics') is specified (i) combinatorially in terms of the propositions in which the word functions (especially absolute propositions) and (ii)

[21]This is in contrast to categorial grammar, where these two aspects are handled jointly by categories which are interpreted (i) combinatorially in the syntax and (ii) as meaning functions in the semantics. A first separation may be found in Montague's PTQ, where the syntactic categories are mapped into corresponding, but nevertheless distinct, semantic types in order to handle phenomena in the area of intensional contexts. See Montague 1974, S. 260f.

conceptually by its perceptionally based prototype (base form) and the associated perceptionally based instantions (tokens).

Because the propositional and the conceptual aspects of meaning are in principle orthogonal to each other, it is possible to build SLIM systems which realize only one of these two aspects of meaning. A SLIM system which handles only the conceptual aspect of single content words is the COLOR READER.[22]

The color reader is a simple robot based on the principle of internal matching pragmatics. It drives around in its task environment and names the the color of the various objects it encounters. For example, faced with a a red box it will say *'red.'* The strength of a purely concept based system is the correct classification of new objects in an open environment (including direct reference).

A SLIM system limited to the propositional aspects of a cognitive agent, on the other hand, is exemplified by LA-base in its initial form. There, the semantic interpretation of sentences consists in the storage of signs, which can be meaningfully interpreted by the user before input and after output, but which are treated as uninterpreted signs inside the database system. The strength of a purely propositional system is the characterization of semantic relations in terms of the composition of content words into propositions according to the semantic principles of functor, argument and modifier.

To extend LA-base into a system which handles also the concepual aspect of meaning, the record structure of the base forms (cf. 3.3.) and the tokens (cf. 3.1.) are each supplemented by an additional attribute (feature) for concepts. This step is illustrated in 5.5. with the example of nouns and applies accordingly to the other record types of LA-base.

5.5. Integration of the conceptual aspect into LA-base

```
G_Noun
base_form  part_of_speech  gender declination_class  P-concept
```

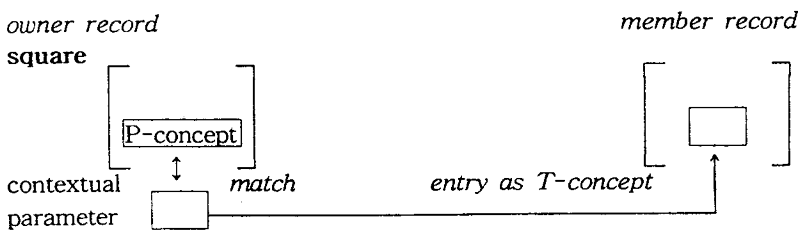
[22]Hausser 1987b; 1989:295 ff. The color reader is a precursor of the robot CURIOUS used in the unpublished work on the SLIM theory of language.

```
T_Noun
index  quantifier case ADJEKTIVE  VERB  T-concept
```

The record structure G_Noun, characterizing the base form of nouns, has the additional attribut P-concept (compare 3.3.), which takes as value a *prototypical* concept. The record structure T_Noun, characterizing tokens of nouns, has the additional attribut T-concept (compare 3.1.), which takes as value a conceptual token instantiating the prototype in the proposition in question.

Assuming that this extended form of LA-base is used as the non-verbal and verbal database component of CURIOUS, then the non-verbal reference to, e.g., a square object found in the task environment would take place as follows:[23]

5.6. Direct, non-verbal reference in LA-base



A square in the visual field of CURIOUS is classified by matching the parameter values (bitmap) with a suitable concept, i.e. the P-concept of 'square.' Then the parameter values of the object are stored as a T-concept in the currently active token of 'square' (instantiation). Thus every encounter with, e.g., a real square is remembered in LA-base by creating a token record with a cognitive copy of the instance.

In addition to the classification of objects, a non-verbal analysis must describe the external environment in terms of propositions, which specify the positions of the objects, their relation to each other, etc. These propositions are generated by the systems as verbal traces during its systematic tracking of the task environment, and have the

[23]Using a definition of the concept 'square' in line with Hausser 1989:314.

same format of distributed, bidirectional, co-indexed tokens as non-verbal propositions originating from language expressions.

The extended version of LA-base functioning in a robot may thus obtain its non-verbal propositions from two different sources: (i) the system's conceptual analysis of its environment and (ii) it's pragmatic interpretation of natural language. Both types of non-verbal propositions serve equally well as parts of the context of use. They differ only in that the tokens of non-verbal propositions originating from language have no value in their T-concept feature.[24]

References

Date, C.J. (1990) *An Introduction to Database Systems*, Fifth Edition, Addison-Wesley Publishing Company.

Elmasri & Navathe (1989) *Fundamentals of Datebase Systems*, Benjamin/Cummings Publishing Company, Inc., Redwood City, CA.

Hausser, R. (1978) *"Surface Compositionality and the Semantics of Mood,"* Groenendijk, Stokhof, and Bartsch (eds.), Amsterdam Papers in Formal Grammar, Vol. II, 1978. Reprinted in Searle, Kiefer, and Bierwisch (eds.), Speech Act Theory and Pragmatics, Reidel, Dordrecht, 1980.

Hausser, R. (1979a) *"A Constructive Approach to Intensional Contexts,"* Language Research, Vol. 18, No. 2, Seoul, Korea, 1982.

Hausser, R. (1979b) *"A New Treatment of Context in Model-Theory,"* Sull' Anaphora, atti del seminario Accademia della Crusca, Florence, Italy, 1981.

Hausser, R. (1981) *"The Place of Pragmatics in Model-Theory,"* Groenendijk et al. (eds.), *Formal Methods in the Study of Language*. Mathematical Center Tracts 135, University of Amsterdam.

Hausser, R. (1983) *"On Vagueness,"* Journal of Semantics 2:273-302.

Hausser, R. (1984) *Surface Compositional Grammar*. Wilhelm Fink Verlag, Munich.

Hausser, R. (1986) *NEWCAT: Parsing Natural Language Using Left-Associative Grammar*. Lecture Notes in Computer Science,

---

[24]When the system hears about, e.g., a read triangle, it may go back within the token sequence of the 'triangle'-lemma to look for an earlier token with a T-concept as a temporary supplement. In this way, the phenomenon of visual imagination in connection with language interpretation may be modeled.

Springer-Verlag, Berlin -- New York.

Hausser, R. (1987) "*Modelltheorie, K unstliche Intelligenz und die Analyse der Wahrheit*," in L. Puntel (ed.) Der Wahrheitsbegriff, Wissenschaftliche Buchgesellschaft, Darmstadt.

Hausser, R. (1989) *Computation of Language*, Springer-Verlag, Symbolic Computation: Artificial Intelligence, Berlin -- New York.

Hausser, R. (1992) "*Complexity in Left-Associative Grammar, Theoretical Computer Science*, Vol. 103, Elsevier.

Montague, R. (1974) *Formal Philosophy*, Yale University Press, New Haven, CT.

Wloka, D.W. (1992) *Roboter Systeme 1*, Springer-Verlag Berlin.

Friedrich-Alexander-Universität Erlangen-Nürnberg
Abteilung Computerlinguistik, Bismarckstraße 12
91054 Erlangen, Germany
email: rrh@linguistik.uni-erlangen.de

접수일자: 1995. 9. 22.
게재결정: 1995. 12. 20.