A Hypothesis on the Origin of the Sign Types

Roland Hausser

Universität Erlangen-Nürnberg
Abteilung Computerlinguistik (CLUE)
rrh@linguistik.uni-erlangen.de

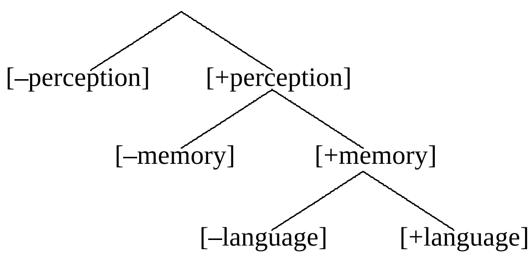
Abstract The functioning of natural language communication depends crucially on the different kinds of signs, i.e., symbol, indexical, and name, and their characteristic mechanisms of reference. In this paper, the meanings of these sign types are traced to cognitive agents with memory but without language.
The argument is based on the task of relating the view-dependent format of a content stemming from an agent's current situation to a view-independent format for storage in memory. The view-dependent analysis of a content, called a task analysis, is built from cognitive structures suitable to serve as the literal meanings of the different sign types. Furthermore, the transfer between the view-dependent format relating to the current situation and the view-independent format of content stored in memory in [–language] agents is shown to be closely related to the alternation between the hearer and the speaker mode in [+language] agents.

1 Introduction

According to Aristotle (Metaphysics, I (A), 980b), all living beings have perception,[1] but only some of them can store what they recognize in memory. The possibility of storing analyzed perception in memory is the precondition for having experience, for learning, and for deliberate action.

Furthermore, living beings with memory may be divided into those with language and those without. By representing these basic distinctions in terms of the binary features [± perception], [± memory], and [± language], we arrive at the following tree structure:

1.1 BASIC DISTINCTIONS CHARACTERIZING COGNITIVE AGENTS



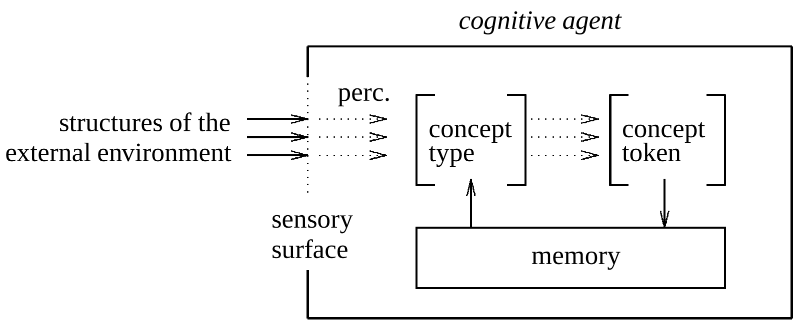
[1] While perception is a necessary condition for life, it is not sufficient. For example, we may grant that robots have perception, yet deny that they are alive.

[–memory] agents must use perception to initiate action directly. From a computational point of view, this kind of reflex-connection between a certain type of perception and an associated action does not require the development of concepts.

[+memory] agents, in contrast, can store what they perceive without performing an external action and they can act without the stimulus of an associated perception. This requires first that perceptions are analyzed in terms of recognitions[2] and stored properly in memory. Second, it requires that intentions are derived and realized as actions.

The transfer of content into the data structure of a [+memory] system during recognition and out of the data structure during action is based on concept types and concept tokens. This is illustrated by the following schematic analysis of recognition:

1.2 CONCEPT TYPES AND CONCEPT TOKENS IN RECOGNITION



Perceptions of the cognitive agent's sensory surface are classified by a suitable concept type provided by memory. Concept types define the necessary properties of a concept by means of constants and the accidental properties by means of variables. For example, the concept type of the geometric object *square* is defined as follows:

1.3 DEFINITION OF THE CONCEPT TYPE OF *square*

$$\begin{bmatrix} \text{edge 1: } \alpha \text{ cm} \\ \text{angle 1/2: } 90^0 \\ \text{edge 2: } \alpha \text{ cm} \\ \text{angle 2/3: } 90^0 \\ \text{edge 3: } \alpha \text{ cm} \\ \text{angle 3/4: } 90^0 \\ \text{edge 4: } \alpha \text{ cm} \\ \text{angle 4/1: } 90^0 \end{bmatrix}$$

[2] What we call a recognition is sometimes called a percept. The latter term does not indicate, however, whether a percept is a raw sensation or a sensation classified in terms of a preexisting concept. Only the latter is called here a recognition.

The necessary properties of this concept type are four angles of 90 degrees and four edges of equal length. The accidental property is the edge length, represented by the variable α. The variable makes the concept type applicable to squares of any size.

When the concept type 1.3 is matched onto the incoming parameter values of 1.2, the variable is bound to a particular edge length, for example 2cm, resulting in the following concept token (which instantiates the concept type 1.3):

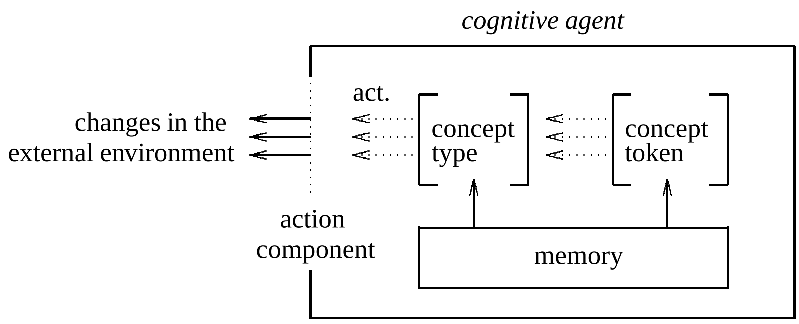
1.4 DEFINITION OF A CONCEPT TOKEN OF *square*

$$\begin{bmatrix} \text{edge 1: 2cm} \\ \text{angle 1/2: } 90^0 \\ \text{edge 2: 2cm} \\ \text{angle 2/3: } 90^0 \\ \text{edge 3: 2cm} \\ \text{angle 3/4: } 90^0 \\ \text{edge 4: 2cm} \\ \text{angle 4/1: } 90^0 \end{bmatrix}$$

This token may be stored in (the episodic part of) the agent's memory.

The inverse of perception and recognition are intention and action. Intention is the process of developing an action cognitively, while action is the mechanism of realizing an intention by changing the external environment.

1.5 CONCEPT TYPES AND CONCEPT TOKENS IN ACTION



The formation of intentions is based on the agent's control structure, current situation, and inferences over content stored in memory. Intentions are represented as constellations of concept tokens and realized as actions by means of corresponding types.

In addition to the definition of concept types and concept tokens, the analysis of contextual (or non-language-based) cognition requires a data structure for indexing and retrieval, an activation of content by means of a time-linear navigation through the database (motor algorithm), a control structure, inferences, etc. (cf. Hausser 2001b).

The computational implementation of perception, recognition, intention, and action is a precondition not only for the construction of [+memory,–language], but also of [+memory,+language] agents. This is because non-language-based recognition and action are an important part of the *context* relative to which language is interpreted.

2 The nonlinguistic nature of the internal context

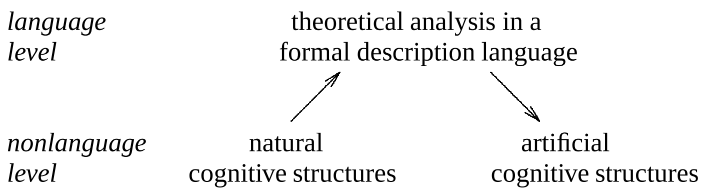
The different aspects of the external world, e.g., the program of a washing machine, the number of planets in the solar system, the atomic structure of the elements, the colors, etc., are inherently nonlinguistic in nature. To call these structures a 'language' is inappropriate because it would stretch the notion of a language beyond recognition.

The essentially nonlinguistic nature of the external originals holds also for their internal representations based on a cognitive agent's recognition. Higher nontalking animals like a dog may well be able to develop something like concept types, to derive concept tokens, to combine them into elementary context propositions, to concatenate these into subcontexts, to draw inferences, and to derive view-dependent analyses of their current situation. These cognitive structures and procedures do not constitute a language, however, because they do not have external surfaces conventionalized in a community and do not serve in inter-agent communication. Instead, they evolve solely as internal, physiologically grown structures.

The contextual structures of a natural [–language] agent, e.g., a dog, acquire a language aspect only if and when they are being *described* by means of a language. Corresponding artificial systems, on the other hand, usually begin with a language-based definition which is then realized in terms of the hardware and software of the implementation. However, even in artificial systems the language aspect may be completely ignored once a system is up and running: on the level of its machine operations (electronic switching) the cognitive procedures of an artificial [–language] agent (nontalking robot) are just as nonlinguistic as those of a natural [–language] agent (dog).

The correlation between the nonlanguage and the language levels in the description of a natural [–language] agent and its artificial model may be described as follows:

2.1 ARTIFICIAL MODELING OF NATURAL COGNITION



The point is that modeling the representation of context within a robot in terms of, for example, propositions based on feature structures and defining the procedures operating on these context structures in terms of a formal grammar are not in conflict with the essentially nonverbal character of these phenomena. Instead feature structures and grammatical algorithms are general abstract formalisms which may be used as much for the description of nonverbal structures as for the description of natural or artificial languages. Furthermore, once the nonverbal structures and the associated inferences have been implemented as electronic procedures they function without any recourse to the language that was used in their construction.[3]

[3] The opposite position is taken by Fodor, who argues for (a) an internal language called 'Mentalese,' but (b) against 'recognitional concepts' like *red*. See Hausser 1999/2000, pp. 64, 65, for an analysis of Fodor's fallacy.

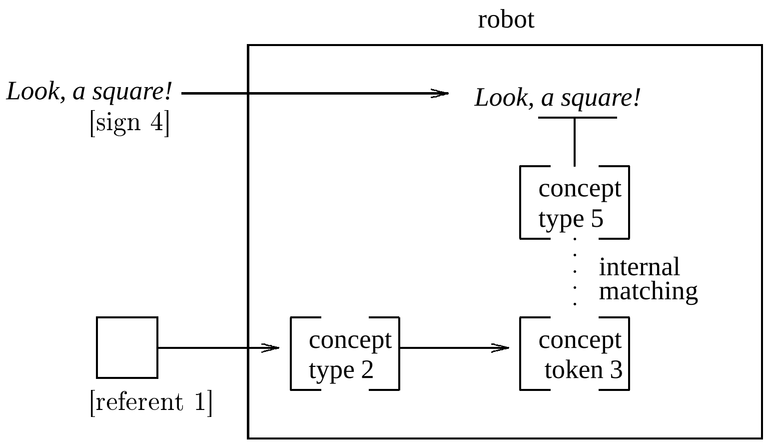
3 Adding language

The primary capability of transferring content into and out of an agent's data structure by means of contextual recognition and action may be complemented by a secondary capability based on language. Language production (speaker mode, export of content) and interpretation (hearer mode, import of content) raise two related questions.

One is a question of coding and decoding: the speaker must code contextual content into natural language, and the hearer must decode natural language into a format resembling contextual content. The other is a question of indexing and retrieval: the speaker must specify in the language sign where in his database the content is coming from (indexing), and the hearer must infer the corresponding position in his database (retrieval) for storing the content correctly in order for communication to be successful.

The SLIM[4] theory of language approaches these questions by first reconstructing immediate reference. Its functioning is illustrated with a sign (square), a referent (geometric object), and an artificial [+language] agent (talking robot in hearer mode):

3.1 IMMEDIATE REFERENCE BASED ON INTERNAL MATCHING



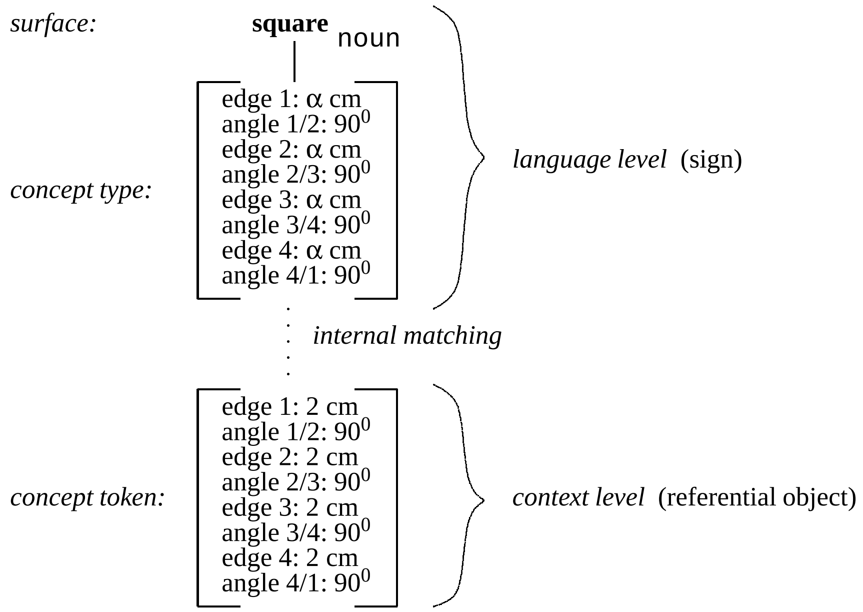
Instead of the usual analysis of reference as a relation between the external sign (4) and the external object of reference (1), reference is reconstructed by SLIM as a purely cognitive process within the cognitive agent. This cognitive process is based on the agent's recognition of the external language sign and of the external referent.

The semantic interpretation of the recognized language sign consists in lexically assigning a literal meaning, defined as a concept type (5) which is identical to the concept type (2). According to this analysis, the evolution of language is based in part on literal meanings which have evolved earlier as the concept types needed for the contextual recognition and action of [–language] agents.

The relation of reference between a sign at the level of language and a recognized object at the level of context is functionally established by the principle of internal matching between a concept type and a concept token:

[4] SLIM (Hausser 1999/2001) stands for the methodological, empirical, ontological, and functional principles of **S**urface compositionality, time-**L**inearity, and **I**nternal **M**atching.

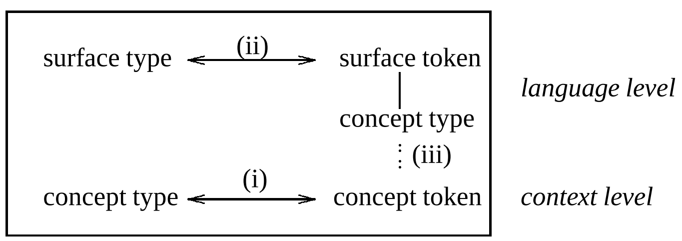
3.2 INTERNAL MATCHING BASED ON THE TYPE-TOKEN CORRELATION



The sign is a fixed lexical structure consisting of (i) a surface (square), (ii) a syntactic category (noun), and (iii) a semantic interpretation defined as a concept type. The referential object is a concept token resulting from recognition. The pragmatics, i.e., the use of the sign in communication, consists in matching the sign's concept type (literal meaning) with a concept token (referential object) provided by the current context of use. SLIM's principle of internal matching (IM) models the flexibility which distinguishes the natural languages from the logical and programming languages.

The evolution of language requires two type-token relations which function in addition to the one characteristic of [–language] agents:

3.3 THREE TYPE-TOKEN RELATIONS IN [+LANGUAGE] AGENTS



The type-token relations arise (i) between concept types and concept tokens during contextual recognition (→) and action (←), (ii) between surface types and surface tokens during surface recognition (→) and synthesis (←), and (iii) between concept types and concept tokens during language interpretation (↓) and language production (↑).

4 Pragmatics

A central concern of semiotics as represented by C.S. Peirce (1839–1914) and C.W. Morris (1903–1979) is pragmatics. This concern is shared by the SLIM theory of language, though with the additional goal of arriving at a computational theory of natural language use in communication.

SLIM formulates pragmatics in terms of seven principles, of which the first relates the literal meaning[5] of language signs, called $meaning_1$, to the speaker meaning of utterances, called $meaning_2$:

4.1 FIRST PRINCIPLE OF PRAGMATICS (POP-1)

> The speaker's utterance $meaning_2$ is the use of the sign's literal $meaning_1$ relative to an internal context.

The crucial notion of use is implemented as an internal matching between literal $meanings_1$ at the level of language and referential objects at the level of context, as illustrated in 3.1 and 3.2 above with the sign type 'symbol'.

The second principle introduces the STAR relative to which a content is coded by the speaker (indexing) and decoded by the hearer (retrieval). The acronym STAR stands for the parameters of **S**pace, **T**ime, **A**uthor, and intended **R**ecipient of the sign.[6]

4.2 SECOND PRINCIPLE OF PRAGMATICS (POP-2)

> A sign's STAR determines the *entry context* of production and interpretation in the contextual database in terms of parameter values.

The STAR is crucial in the case of mediated reference, i.e., when the speaker refers to a context which is removed, for example, spatio-temporally, from the circumstances of utterance or when the hearer refers to a context which is removed from the circumstances of interpretation.

The third principle describes how complex signs (sequences of word forms in sentences or texts) are related to corresponding items at the level of context.

4.3 THIRD PRINCIPLE OF PRAGMATICS (POP-3)

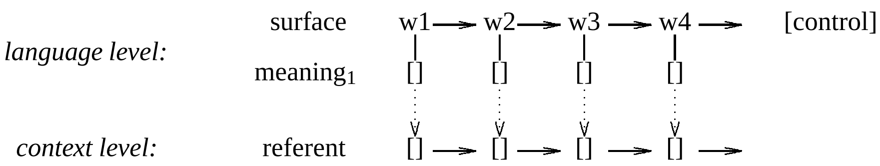
> The matching of the signs' $meaning_1$ with corresponding items at the level of context is incremental, whereby in production the elementary signs follow the time-linear order of the underlying thought path, while in interpretation the thought path follows the time-linear order of the incoming elementary signs.

In language interpretation, the navigation through the context is controlled by the language signs: the hearer follows the surfaces of the signs, looks up their $meanings_1$ in the lexicon, and matches them with suitable referents at the level of context:

[5] SLIM's use of literal meanings is in contrast to traditional semiotics, especially Morris. The aims of SLIM in comparison to other theories of language are discussed in Section 6 below.

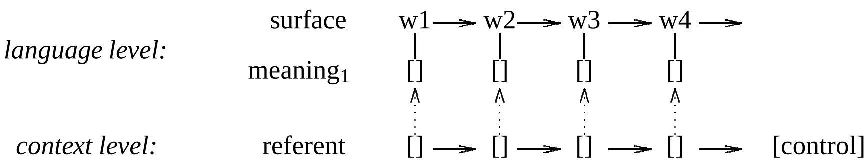
[6] The STAR is an extension of the 'point of speech' by Reichenbach 1947, p. 288, which refers to the temporal parameter T only and is terminologically restricted to spoken language. The STAR was first described in Hausser 1989, pp. 274 f. See also Hausser 1999/2001, pp. 93 f.

4.4 Schema of language interpretation (hearer mode)



In language production, the navigation control is located in the context, i.e., the data structure of the speaker's memory. Each unit traversed at the level of context is matched with the corresponding meaning$_1$ of a suitable word form:

4.5 Schema of language production (speaker mode)



Utterance of the word form surfaces allows the hearer to reconstruct the speaker's time-linear navigation path.

The schemata 4.4 and 4.5 agree with the natural view that interpretation (↓) and production (↑) are inverse[7] vertical procedures. Nevertheless, interpretation and production have their main direction in common, namely a horizontal time-linear structure (→) – in line with de Saussure's second law. The time-linear syntactic-semantic interpretation and contextual navigation are based on the algorithm of LA-grammar (Hausser 1992).

5 Different kinds of signs

The basic setup of pragmatic interpretation has been illustrated in 3.2 with a specific sign type, called *symbol*. Other sign types of natural language are *indexicals* like now, here, or this, and *proper names* like John or R2D2. In addition, there is the sign type of *icons*, which is marginal for synchronic natural language communication,[8] but important for explaining the evolution of symbols.

In modern times, the theory of signs was founded by Peirce, who analyzes the sign types symbol, indexical, and icon, but omits names. Symbols are defined as follows:

> A symbol is a sign which would lose the character which renders it a sign if there were no interpretant. Such is any utterance of speech which signifies what it does only by virtue of its being understood to have signification.
>
> Peirce 1940, p. 104.

---

[7] This view is expressed, for example, by Mel'čuk 1988, p. 50.

[8] As pointed out by de Saussure 1967, pp. 81, 82. See Hausser 1999/2001, pp. 114 f.

Similarly, an index is defined as a sign which would lose the character which renders it a sign as soon as the object it is pointing at is removed; an icon is defined as a sign which retains its character as a sign even if there is no object to refer to, and no interpretant.

The disadvantage of Peirce's definitions is that they are unsuitable for computational implementation. Alternatively, SLIM explains the functioning of the different sign types, i.e., their respective mechanisms of reference, in terms of their cognitive structure. This analysis is formalized as the principles of pragmatics PoP-4 to PoP-6.

5.1 FOURTH PRINCIPLE OF PRAGMATICS (POP-4)

> The reference mechanism of the sign type **symbol** is based on a meaning$_1$ which is defined as a concept type. Symbols refer from their place in a positioned sentence by matching their meaning$_1$ with suitable contextual referents.

The reference mechanism of symbols is called iconic, because the functioning of symbols and icons is similar: both can be used to refer spontaneously to new objects of a known kind. The difference resides in the relation between the meaning$_1$ and the surface, which is arbitrary in the case of symbols, but motivated in the case of icons.

5.2 FIFTH PRINCIPLE OF PRAGMATICS (POP-5)

> The reference mechanism of the sign type **indexical** is based on a meaning$_1$ which is defined as a pointer. An indexical refers by pointing from its place in the positioned sentence to a value in an appropriate parameter.

Indexical reference is illustrated by the adverbs here and now, which point to values in the spatial and temporal parameters of an utterance, respectively, and the pronouns I and you, which point to the author and the intended recipient, respectively.

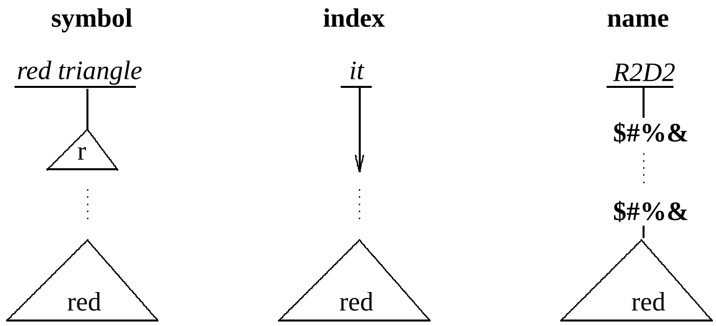
5.3 SIXTH PRINCIPLE OF PRAGMATICS (POP-6)

> The reference mechanism of the sign type **name** is based on a meaning$_1$ defined as a private marker which corresponds to a private marker contained in the cognitive representation of the corresponding referential object. Reference with a name consists in matching the two private markers.[9]

As an example, consider agent A observing a dog. For easier reference, the cognitive structure representing the dog in agent A's context is abbreviated by the private marker $#%&. Later, another agent calls the dog by the name Fido. Agent A adopts this name by attaching the private marker $#%& to the public surface Fido. Henceforth, the name Fido refers for A to the dog in question by matching the private marker attached to the name with the corresponding marker attached to the referent.

The respective structural basis of iconic, indexical, and name-based reference is illustrated in the following schematic comparison in which the three sign types are used to refer to the same contextual object, i.e., a red triangle.

[9] In analytic philosophy, names have long been a puzzle. Attempts to explain their functioning range from *causal chains* to *rigid designators* (cf. S. Kripke 1972). The present analysis is more general than the one in Hausser 1999/2001.



All three sign types have a meaning_1 which is firmly attached to a surface. The symbolic expression red triangle refers on the basis of the type-token relation: the meaning_1 is a concept type which matches a corresponding concept token in a limited range of contextual candidates. The indexical it refers on the basis of pointing: the meaning_1 is a characteristic pointer which points at the referential object within an associated parameter (here, third person, i.e., everything that is neither author nor addressee). The name R2D2 refers on the basis of matching private markers: the meaning_1 is $#%&, which is matched with an identical marker in the contextual referent.

All three mechanisms of reference must be analyzed as internal, cognitive procedures because it would be ontologically unjustifiable to locate the fixed connections between their signs' surface and meaning_1 in external reality. Instead, meaning_1 is assigned cognitively by means of a lexicon which all speakers-hearers have to learn.

For explaining the phylo- and ontogenetic development of natural language it is of interest that the basic mechanisms of iconic and indexical reference[10] constitute the foundation of nonverbal and preverbal communication as well. Thereby

1. nonverbal iconic reference consists in spontaneously imitating the referent, and
2. nonverbal indexical reference consists in pointing at the referent.

While essentially limited to face-to-face communication, these nonverbal mechanisms of reference may be quite effective. By avoiding the use of conventionally established surfaces, nonverbal reference allows spontaneous communication in situations in which no common language is available.

It is important to note that the distinction between the different *sign types*, i.e., symbol, indexical, and name, is orthogonal to the distinction between the main *parts of speech*, i.e., noun, verb, and adjective, as well as to the corresponding distinction between the basic *elements of propositions*, i.e., argument, functor, and modifier.

5.5 SEVENTH PRINCIPLE OF PRAGMATICS (POP-7)

> The sign type *symbol* occurs as noun, verb, and adjective. The sign type *indexical* occurs as noun and adjective. The sign type *name* occurs only as noun.

[10] The early form of name-based reference is not included here because it constitutes the transition to language-based communication. See Hausser 1999/2001 for comparison.

The orthogonal correlation between sign types and parts of speech described in PoP-7 may be illustrated graphically as follows:

5.6 RELATION BETWEEN SIGN TYPES AND PARTS OF SPEECH

name	*Peter*		
indexical	*this*	*now*	
symbol	*triangle*	*red*	*contain*
	noun	adjective	verb

The sign type which is the most general with respect to different parts of speech is the symbol, while the name is the most restricted. Conversely, the part of speech (and, correspondingly, the propositional element) which is the most general with respect to different sign types is the noun (object), while the verb (relation) is the most restricted.

6 The role of context in communication

The SLIM-theoretical analysis of natural language aims at a functional model of communication. For functioning in the real world, the model has to be procedural rather than metalanguage-based.

A procedural model must have a declarative specification defining its necessary properties. In contradistinction to a meta-language based approach, however, it is not dependent on set theory to provide immediately obvious basic meanings.

Instead, basic meanings are programmed as concept types for classifying incoming parameter values, such as colors or geometric shapes, and for realizing outgoing intention tokens, such as certain actions of locomotion or gripping. Furthermore, the agent's saying something true is characterized in terms of contextual recognition and action, as well as language interpretation and production, working properly.

SLIM differs from previous theories of language because it provides an objectively testable method of verification. This method consists in building artificial cognitive agents which can recognize new objects of a known kind in their real world environment, talk about which objects they found in the past or which objects they hope to find in the future, understand questions and commands regarding their experiences and actions, etc.[11]

For this, the different sign types must be analyzed in terms of their cognitive structure and associated functioning – in contradistinction to the classificational approach of Peirce. As we have seen, Peirce distinguishes between symbols, indices, and icons in terms of whether or not there has to be an interpretant, and whether or not there has to be a referent, in order for the respective types of signs 'to retain their character.'

[11] For a simple 'fragment' comprising language understanding, conceptualization, language production, inferencing, and querying, see Hausser 2001b.

The realization of SLIM in terms of a functioning artificial agent cannot prove that this theory is the only one correct. However, given alternative theories, the one functioning best is to be preferred – provided the theoretical goal is a computational model of how communication works and the practical goal comprises unrestricted human-computer communication. Note that previous theories of language, such as pragmatism, structuralism, behaviorism, model theory, speech act theory, or nativism, have not been designed for these goals. Not surprisingly, they are unsuitable for reaching them.

In addition to functional performance, SLIM may be supported by other desiderata of scientific research, such as compatibility with results from psychological experiments, findings of neurology, or a plausible explanation of how [+language] agents have evolved from [–language] agents.[12] Regarding the latter, SLIM is special in comparison to previous theories of language in that it begins with an explicit definition of the cognitive agent's internal context. This is motivated as follows:

First, modeling the agent's context as a database is functionally necessary for realizing certain communication types,[13] such as language-controlled action (telling the robot what to do) and commented recognition (the robot describing what it sees). It is also necessary in mediated reference, when content independent of the current task environment (for example, regarding past events) is being read into and out of the contextual database by means of natural language.

Second, starting with modeling the context in [–language] agents and then building the language level on top of it is in concord with evolution. Thereby the development of concept types needed for contextual recognition and action in [–language] agents (cf. Section 1) provides cognitive structures suitable to serve as the meaning_1 of symbols in [+language] agents: the evolution of this sign type merely requires *reusing* already available concept types by attaching them to categorized surfaces (cf. Section 3).

The question now is whether the evolution of the other sign types can be described in a similar manner: do [–language] agents have a need to develop the indexical pointers and the private name markers, such that these cognitive structures may be simply reused in the evolution of the corresponding sign types by attaching them to categorized surfaces?

7 Relating stored content to the current situation

From the viewpoint of evolution it seems plausible that the meaning_1 of the different sign types develops already in [–language] agents. Our argument for this, however, is functional in nature. It is based on the computationally motivated hypothesis that [–language] agents need a simplified, purpose-oriented view which is superimposed on the many details of their current recognitions and intentions.

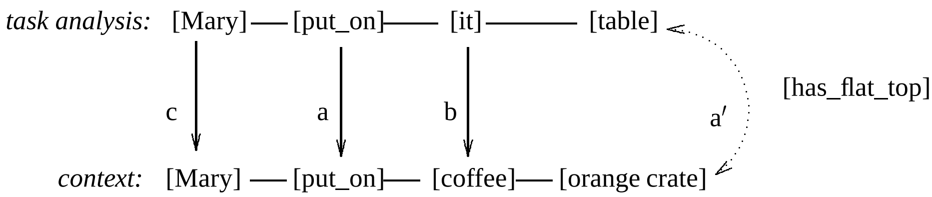
Such a view-dependent representation is called a *task analysis*. This momentary construction exists in addition to and simultaneously with the corresponding context. While the context constitutes the content's literal representation, the task analysis *refers* to the content by using indexical pointers and metaphorically used concepts in addition to literally used concepts and private markers.

[12] A major flaw of Grice's 1965 theory of sentence meaning and utterer's meaning is that it cannot explain how types initially evolved – as pointed out, for example, by Searle 1969, p. 44f.

[13] They are classified as SLIM-1 – SLIM-10 in Hausser 1999/2001, pp. 469–473.

Consider the following example:

7.1 PRIMARY TASK ANALYSIS OF AN IMMEDIATE CONTEXT



The above context is assumed to be a complex situation consisting of many facts. Most of them are omitted in the representation, however, for the sake of simplicity, for example, the shape of the room, the color of the walls, the position of the windows, etc. From these, a small subset, represented as the proposition *Mary put_on coffee orange-crate*, is selected by the task analysis, thus providing a simplified, purpose-oriented extract.[14]

The task analysis is a secondary representation which selects corresponding referents at the level of context by means of concept types (a), indexical pointers (b), and private markers (c). The selection is guided by the cognitive agent's habits, knowns, unknowns, likes, dislikes, needs, and purposes. The latter may even lead to viewing, for example, an orange crate metaphorically as a table (a′), based on the property *has flat top*. The simultaneous two-level representation has the following functions:

First, for recognition and action, the task analysis complements the current immediate context with a simplified, purpose-oriented representation which highlights relevant and/or familiar patterns.[15] This simplified representation helps to keep track of referents when interacting with the task environment in a sequence of recognitions and actions.

Second, for storage in memory, the task analysis selects relevant aspects from the current immediate context to avoid overflow. When the selected content is stored in memory, however, the task analysis' view-dependent indexical and metaphorical aspects must be eliminated. This requires a mapping from the view-dependent task analysis to a view-independent representation suitable for long-term storage.

Third, for relating stored view-independent content to the current situation, a *secondary* task analysis is constructed from the viewpoint of the cognitive agent's current tasks and purposes. This requires a mapping from the view-independent representation of long-term storage to a representation which takes the agent's current viewpoint into account.

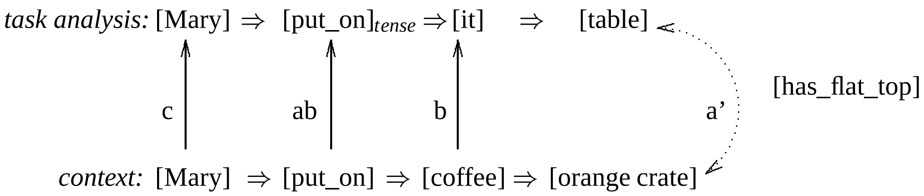
Consider the following example:

---

[14] For better readability, the private name marker is represented as 'Mary' in 7.1 rather than $#%&, the concepts as 'put_on' and 'table' rather than explicit concept types at the level of the task analysis and concept tokens at the level of context, and the indexical pointer as 'it' rather than an explicit vector pointing at a certain value of a certain parameter.

[15] The simultaneous evolution and separation of the levels of context and task environment seems to have occurred quite gradually. In a frog, for example, the level of context impinges on a primitive task analysis realized as a prewired array of reflexes, such as *small moving spot–jump for it* or *larger shadow–go for cover*.

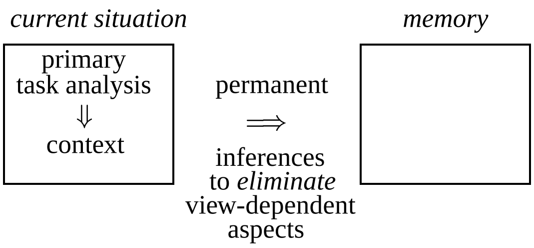
7.2 SECONDARY TASK ANALYSIS OF A STORED CONTENT



The proposition *Mary put_on coffee orange-crate* has been stored in memory for some time. When the proposition is activated by means of navigating through it (cf. context), a secondary task analysis is constructed. It expresses the proposition's temporal relation to the current situation indexically by assigning tense to the verb (ab). Furthermore, assuming that the referent *coffee* has already been activated, the secondary task analysis represents it indexically (b). Like a primary task analysis, a secondary task analysis may also refer metaphorically, such as viewing the orange-crate as a table (a′).

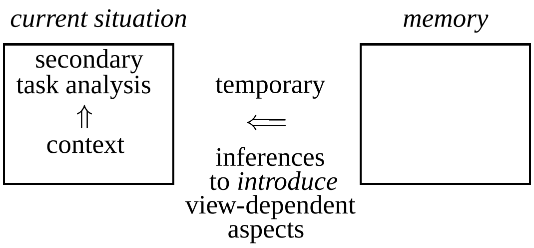
When the view-dependent selection from a current content provided by a primary task analysis is read into memory, it must be modified into a view-independent format by way of inferences. The result is permanent insofar as the view-independent content stored in memory may remain unchanged over time.

7.3 STORAGE IN MEMORY



Conversely, when a content stored in memory is adapted to the current situation, its view-independent format is complemented by a view-dependent secondary task analysis, based on inferences. The result is temporary insofar as (i) the current situation may change quickly and (ii) the same content may be adapted to different current situations.

7.4 RETRIEVAL FROM MEMORY



Formalized examples of the inferences eliminating and introducing view-dependent aspects are presented in Hausser 1999/2001.[16]

8 Conclusion

The derivation of task analyses for (i) interacting with the current situation, (ii) storing current interactions in memory, and (iii) adapting stored content to the current viewpoint in [–language] agents prepares the evolution of natural language as follows:

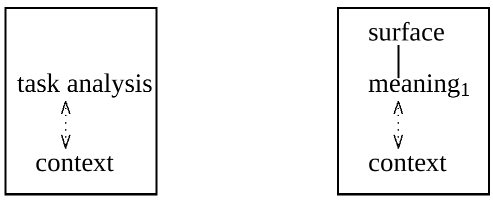
1. The elements from which task analyses are built are suitable to be reused as the meaning_1 of symbols, indexicals, and names such that the emergence of these sign types requires no more than attaching the preexisting meanings_1 to public surfaces.
2. The inferences eliminating the view-dependent aspects of a primary task analysis for the purpose of long-term storage are essential also for the hearer mode of natural language interpretation.
3. The inferences guiding the choice of elements in the construction of a view-dependent secondary task analysis, e.g., a nonliteral use of a concept type, a pointer, or a private marker, are needed also in the speaker mode of natural language production.

More generally, the [–language] agents' need to (i) transform view-dependent content into a view-independent format suitable for storage and (ii) adapting view-independent content to a view of the current situation has a direct counterpart in [+language] agents, namely the switching between the hearer mode and the speaker mode.

Consider the following example: A [–language] agent stores at some point in time the sequence of propositions *I am visiting the Munchkins. The Munchkins serve coffee. Mary puts coffee on orange crate.* When this content is activated one year later in memory, it may be adapted to the present situation by the following task analysis: *I visited the Munchkins last year. They served coffee. Mary put_on it table.*

The point is that a [–language] agent's cognitive adaption of the stored content to the present situation requires the same modifications as a [+language] agent's corresponding coding (speaker mode) into natural language. The only difference is the absence vs. presence of language surfaces, as shown below:

8.1 COMPARING [–LANGUAGE] AND [+LANGUAGE] AGENTS



[+memory, –language] [+memory, +language]

[16] For inferences eliminating view-dependent aspects see p. 499, 24.4.4 and 24.4.6, for inferences introducing view-dependent aspects see p. 495, 24.4.3, and p. 499, 24.4.5.

The analogous situation holds for the opposite direction of eliminating view-dependent aspects, which corresponds to the operations necessary for the hearer's interpretation.[17]

In summary, the crucial qualitative contribution of language over and above the cognitive functions of [–language] agents consists in the added ability of *communicating* content from one agent to another by using the signs' public surfaces. It goes without saying that this added qualitative ability has enormous quantitative consequences with regard to the amount of knowledge potentially available to [+language] agents.

References

Davidson, D. & G. Harman (eds.) (1972) *Semantics of Natural Language*, D.Reidel, Dordrecht.

Fodor, J. (2000) *In Critical Condition, Polemical Essays on Cognitive Science and the Philosophy of Mind*, Bradford Books.

Grice, P. (1965) "Utterer's Meaning, Sentence Meaning, and Word Meaning," *Foundations of Language*, 4:1–18.

Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283-308, Elsevier.

Hausser, R. (2001a) "Spatio-Temporal Indexing in Database Semantics," in A. Gelbukh (ed.) *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science Vol. 2004. Springer-Verlag, Berlin, New York.

Hausser, R. (2001b) "Database Semantics for Natural Language," *Artificial Intelligence*, Vol. 130.1:27–74, Elsevier.

Hausser, R. (1999/2001) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language*, Springer-Verlag, Berlin, New York, 2nd Ed.

Kripke, S. (1972) "Naming and Necessity," in Davidson & Harmann (eds.), 253–355.

Mel'čuk, I. A. (1988) *Dependency Syntax: Theory and Practice*, State University of New York Press, New York.

Morris, C.W. (1946) *Signs, Language, and Behavior*, New York.

Peirce, C.S. (1940) *The Philosophy of Peirce: Selected Writings*. J. Buchler (ed.), London.

Reichenbach, H. (1947) *Elements of Symbolic Logic*, The Free Press, New York.

Saussure, F. de (1972) *Cours de linguistique générale*, Édition critique préparée par Tullio de Mauro, Éditions Payot, Paris.

Searle, J.R. (1969) *Speech Acts*, CUP, Cambridge, England.

[17] An important special case requiring inferences beyond those needed for introducing and eliminating task analyses is the interpretation of permanent natural language signs, e.g., a letter. For a detailed analysis see Hausser 2001a, Section 7, or Hausser 1999/2001, Section 22.5.