

RESEARCH

Open Access



A computational treatment of generalized reference

Roland Hausser*

*Correspondence:
rrh@linguistik.uni-erlangen.de
Universität Erlangen-
Nürnberg (em.), Erlangen,
Germany

Abstract

Purpose: Medieval logic defined reference as a relation between language and objects in the world. Recently, however, the term “representational token” has been used instead of language (Reimer and Michaelson in *The Stanford encyclopedia of philosophy*: winter 2014 edition. <http://plato.stanford.edu/entries/reference/>, 2014). This allows for reference with and without language. In a similar vein, Database Semantics (DBS) has implemented concept-based reference as a matching between two contents. If a content is attached to a language surface it is called the literal meaning₁ of the surface.

Methods: Referring with a content (as a representational token), regardless of whether or not it is attached to a surface, leads to a generalized notion of reference. An example of reference without language is identifying a current nonlanguage recognition with something seen before. Another example is identifying a nonlanguage recognition with an earlier language content, e.g. something read (for example, in a guide book) or heard about.

Results: In addition to the concept-based reference mechanism of (i) symbols [We follow the terminology used by Peirce (CP 2.228, 2.229, 5.473) for his theory of signs.] (“[Reference by matching \(symbol\)](#)” section), natural language uses the reference mechanisms of (ii) indexicals (“[Reference by pointing \(indexical\)](#)” section) based on pointers, and of (iii) names (“[Reference by baptism \(name\)](#)” section) based on acts of generalized baptism. A fourth kind of reference is co-reference (“[Reference by address \(coreference\)](#)” section), based on identity implemented by means of an address; it occurs as a variant of referring with indexicals and symbols, and is the foundation of name-based reference.

Conclusions: This paper systematically reconstructs the mechanisms of reference as they function with and without language in an agent-based computational framework. Language-dependent surfaces play a role only in the automatic word form recognition of the hear mode and the automatic word form production of the speak mode. In conclusion, the agent-based reconstruction of reference is applied to the medieval distinction between *de dicto* and *de re* (“[De dicto/de re](#)” section).

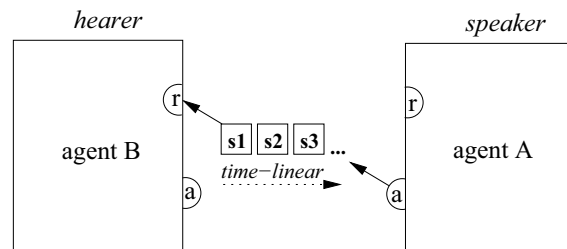
Keywords: Reference based on matching, Pointing, Baptism and identity, Hear mode, Speak mode, Time-linear derivation order, Talking robot

Agent-based ontology

When two agents communicate with each other by means of a natural language, the speaker uses its external action interface to produce a sequence of language surfaces while the hearer uses its external recognition interface to identify the elements of the sequence. The sequence is time-linear in the sense that it is linear like time and in the

direction of time. In accordance with the Western writing convention, the progression of time is shown in the direction from left to right.

Physical framework of communication



The recognition and action interfaces of the agents are indicated by half circles marked with r and a, respectively. The language surfaces are represented by boxes containing s1, s2, s3...,. As agent-external modality-dependent sound waves (speech), dots on paper (writing), or gestures (signing), the surfaces may be measured and described with the methods of the natural sciences, but have no meaning and no grammatical properties whatsoever.

The first surface leaving the speaker is the first to reach the hearer. The last surface leaving the speaker is the last to reach the hearer.¹ All other aspects of language communication are agent-internal, modality²-independent, and cognitive.

Modality-independence may be illustrated by the basic operations of arithmetic, i.e., addition, subtraction, multiplication, and division. They exist at a level of abstraction which may be realized equivalently as the operations (1) of a human, (2) a mechanical calculator, or (3) an electronic computer.³

With autonomous robots still absent in today's computational linguistics, the external framework "Physical framework of communication" may be simulated, using the keyboard and the screen of standard computers as primitive recognition and action components. This, however, works only for the transfer of surfaces. It does not work for nonlanguage recognition and action, which are required for a cognitive reconstruction of reference. For example, the agent's ability to refer to agent-external items is needed for fulfilling a request like *Pick up the blue square!* or to report how many blue squares there are in the agent's current task environment.

Elementary concepts

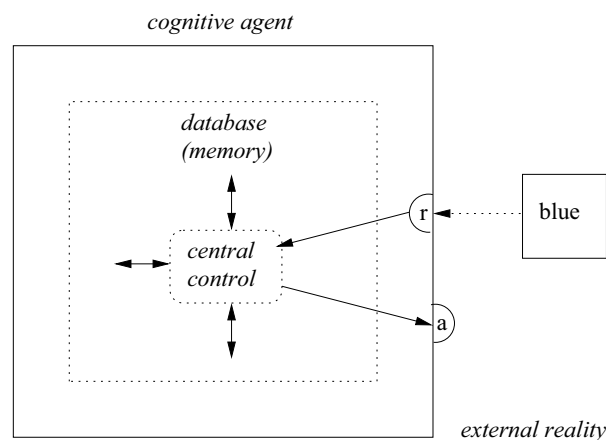
The minimum in reconstructing higher-level cognition is (1) an agent-internal memory, (2) a central control embedded into and interacting with memory, (3) a mapping from the recognition interface to central control, and (4) a mapping from central control to the action interface. Consider an agent recognizing the non-language object *blue square*:

¹ In "Physical framework of communication", "Natural language transfer mechanism", and "Immediate reference as a purely cognitive procedure" this is expressed graphically by placing the hearer to the left and the speaker to the right. If the order is reversed, the progression of time would have to shown from right to left.

² The term *modality* is being used in several different fields of science. As employed here, modality is known as *sensory* modality (Chen 2006, Sect. 6.13.1)

³ According to (Wiener 1961, p. 132): "Information is information, and not matter or energy."

Basic structure of cognition



The external interfaces of recognition and action constitute the agent's *peripheral cognition* ("[Conceptual view of interfaces and components](#)"). In recognition, peripheral cognition maps modality-dependent raw data into modality-independent agent-internal concepts. In action, it maps modality-independent blue-prints into modality-dependent external raw data. This provides the *grounding* (Barsalou 2008) of Database Semantics in the recognition and action of natural or artificial cognitive agents.

The mappings between modality-dependent raw data and modality-independent concepts are formally based on the type-token distinction, familiar from philosophy.⁴ The type of a concept describes the *necessary* properties, while an associated token is an instantiation with certain additional *accidental*⁵ properties. As an example consider the recognition of colors (Hausser 1989, p. 296 ff). In physics, they are defined as intervals on the one-dimensional scales of electromagnetic wave length and frequency. Accordingly, the type and a token of the color blue may shown as follows:

Type and token of the color called blue

type	token
$\begin{bmatrix} \text{wavelength: } \alpha \\ \text{frequency: } \beta \end{bmatrix}$ <p>where $\alpha \in 490\text{--}450\text{ nm}$ and $\beta \in 610\text{--}670\text{ THz}$</p>	$\begin{bmatrix} \text{wavelength: } 470\text{ nm} \\ \text{frequency: } 640\text{ THz} \end{bmatrix}$

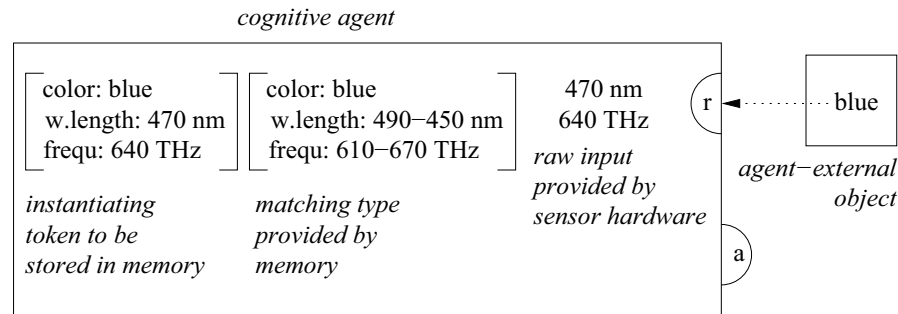
The type specifies the wavelength and the frequency of the color *blue* by means of variables which are restricted to the corresponding intervals provided by physics. The token uses constants which lie within these intervals.

⁴ The type-token distinction was introduced by Peirce (CP 4: 537).

⁵ The term accidental is used here in the philosophical tradition of Aristotle (Barnes 1974), who distinguishes in his *Metaphysics*, Books ζ and η , between the necessary and the accidental (incidental or coincidental—*kata sumbebēkos*) properties of an object in nature.

In the recognition of colors, the type provided by the agent's memory and the raw input data provided by a sensor interact as follows, resulting in a classified token:

Type and token in color recognition



A sensor measures the wavelength 470 nm and frequency 640 THz in an agent-external object. These values lie within the intervals 490–450 nm and 610–6700 THz of the color blue and thus match the type. In the instantiating token, the wavelength and frequency intervals of the type are replaced by the measured values. The feature structures representing concept types and tokens may be extended as needed, for example, with an additional attribute for color intensity.

Next consider the type and the token of a two-dimensional geometric object:

Type and token of the concept square

<i>type</i>	<i>token</i>
$\left[\begin{array}{l} \text{edge 1: } \alpha \text{ cm} \\ \text{angle 1/2: } 90^\circ \\ \text{edge 2: } \alpha \text{ cm} \\ \text{angle 2/3: } 90^\circ \\ \text{edge 3: } \alpha \text{ cm} \\ \text{angle 3/4: } 90^\circ \\ \text{edge 4: } \alpha \text{ cm} \\ \text{angle 4/1: } 90^\circ \end{array} \right]$	$\left[\begin{array}{l} \text{edge 1: 2 cm} \\ \text{angle 1/2: } 90^\circ \\ \text{edge 2: 2 cm} \\ \text{angle 2/3: } 90^\circ \\ \text{edge 3: 2 cm} \\ \text{angle 3/4: } 90^\circ \\ \text{edge 4: 2 cm} \\ \text{angle 4/1: } 90^\circ \end{array} \right]$

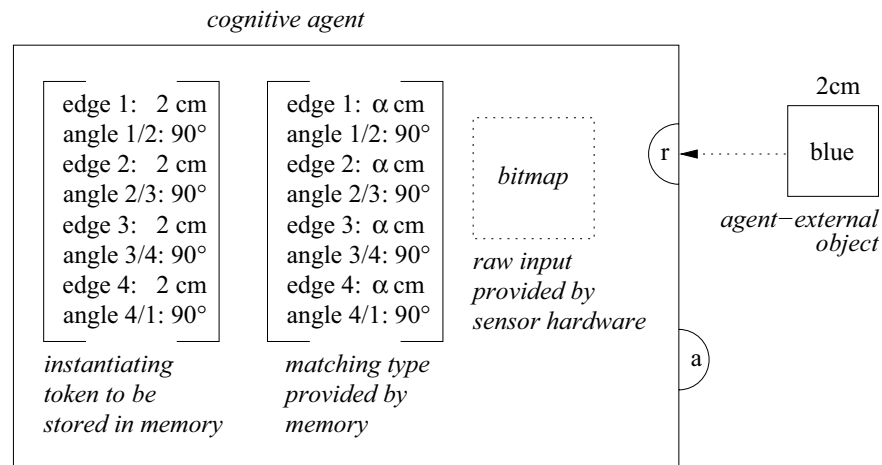
where α is a length

Here, the type and the token share attributes which specify (1) the number of equally long edges and (2) the angle of their intersections. The type and the token differ only in their edge lengths. The latter is accidental in that the type matches an infinite number of square tokens with different edge lengths.⁶

In analogy to “[Type and token in color recognition](#)”, recognition of a square may be shown as follows:

⁶ In an artificial agent, the type may be implemented as a pattern-matching software which recognizes tokens by approximating raw bitmap outlines (Hausser 1999, Sect. 3.2.1).

Type and token in recognizing a square



The type matches the outline of all kinds of different squares, whereby its variables are instantiated in the resulting tokens.

Today, there exist pattern recognition programs which are already quite good at recognizing geometric objects.⁷ They differ from our approach in that they are based almost completely on statistics. However, even if the terms of the type and the token may not be found in their theoretical descriptions, the type-token distinction is nevertheless implicit in any pattern recognition processing. Furthermore, the rule-based, incremental procedures⁸ of pattern recognition presented in Hausser (2005) are well-suited to be combined with statistical methods.⁹

The elementary concepts of nonlanguage recognition are complemented by those of action. For example, the concept *take* is defined as the type of a gripping action which is instantiated as a token to be realized as raw data. The token differs from the type in that it is adapted to a specific gripping occasion. It holds in general for recognition that raw data are classified by a type and instantiated as a token, while in action a type is specialized into a token which is passed to a suitable action component for realization as raw data (Hausser 1999, Sect. 3.3.5).

The interaction between the agent's external interfaces, the types, the tokens, and the memory must be hand in glove. For example, if the agent has no sensor for measuring electromagnetic wavelength/frequency, colors cannot be recognized—even if the proper types were available from memory. Conversely, without the types the raw data provided by a suitable sensor cannot be classified and instantiated as tokens. Also, without a memory the types cannot be provided for recognition and action, and the tokens cannot be stored.

⁷ As shown by the work of Steels (1999), suitable algorithms may evolve new types automatically from similar data by abstracting from what they take to be accidental.

⁸ They are based on an incremental, memory-based procedure of pattern recognition using geons (Biederman 1987).

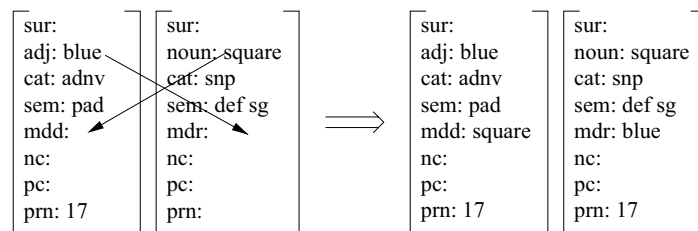
⁹ For building a talking robot, the automatic evolution of types has to result in concepts which correspond to those of the intended language community. This may be achieved by presenting the artificial agent with properly selected data in combination with human guidance (guided patterns method, Hausser 2011, Sect. 6.2).

Data structure and database schema

The concepts defined in “[Type and token of the color called blue](#)” and “[Type and token of the concept square](#)” constitute elementary cognitive contents, but they do not provide any means for being connected, as in *blue_square*. For this, DBS lexically embeds the concepts as *core values* into nonrecursive¹⁰ feature structures with ordered attributes, called *proplets* (because they are the elementary building blocks of propositions, in analogy to droplet). A feature structure is built from features. In computer science, a feature is defined as an attribute-value pair (avp), e.g. [noun: square], with noun: as the attribute and square as the value.

The embedding of core values into proplets allows their concatenation by means of value copying. For example, the proplets *blue* and *square* may be connected into the content of *blue_square* as follows:¹¹

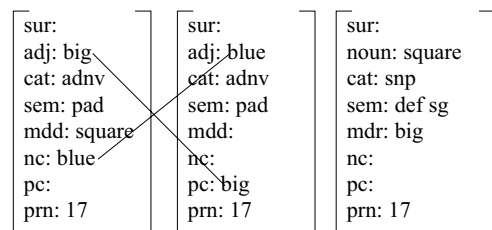
Concatenation by cross-copying



The nature of the semantic relation between *blue* and *square* is characterized by the attributes mdr (modifier) and mdd (modified). The relation is implemented by copying the core value of *square* into the mdd slot of *blue* and the core value of *blue* into the mdr slot *square*. In addition, the prn value of *blue*, here 17, is copied into the prn slot of the next word proplet *square*.

Next consider extending “[Concatenation by cross-copying](#)” to an intrapropositional coordination:

Coordination in *big blue square*



The relation of intrapropositional coordination is coded by the nc (next conjunct) and pc (previous conjunct) attributes of the conjoined adjectives.

¹⁰ A feature structure is nonrecursive if there is no recursive embedding of feature structures as values. Recursive feature structures are unsuitable for (1) contents with a coordination structure, (2) the pattern matching needed for (a) modeling reference and (b) applying operation patterns to input, and (3) storage and retrieval in a database. Unordered attributes are inefficient for computers and humans alike. Recursive feature structures with unordered attributes are not used in DBS.

¹¹ The algorithm used for connecting (hear mode) and activating (think mode) proplets is time-linear Left-Associative Grammar (Hausser 1992). The sur attribute takes the language dependent surface as value. For a detailed description of the attributes and values used in proplets for describing English see Hausser (2006; Appendix A3).

The diagonal lines in “[Coordination in big blue square](#)” are intended as optical support for the reader. Technically, however, they are redundant and may be omitted. The real method of establishing semantic relations in DBS is by addresses coded declaratively as values and implemented procedurally as pointers. This method makes the proplets forming a complex content order-free, allowing the database to store them independently of the semantic relations between them.

For example, no matter where the storage mechanism of the database puts the adnominal *big*, its modified may be found via the primary key consisting of the mdd value square and the prn value 17. Similarly, no matter where the noun *square* is stored, its modifier may be found via the mdr value big and the prn value 17. And accordingly for the intrapropositional coordination in “[Coordination in big blue square](#)”.

As another example consider the content of Julia knows John., represented as the following set of connected proplets:

Content of Julia knows John. as a set of proplets

$\left[\begin{array}{l} \text{sur: Julia} \\ \text{noun: Julia} \\ \text{cat: snp} \\ \text{fnc: know} \\ \text{prn: 625} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: knows} \\ \text{verb: know} \\ \text{cat: \#s3' \#a' decl} \\ \text{arg: Julia John} \\ \text{prn: 625} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: John} \\ \text{noun: John} \\ \text{cat: snp} \\ \text{fnc: know} \\ \text{prn: 625} \end{array} \right]$
---	--	---

The simplified proplets are held together by a common prn value, here 625. The functor-argument is coded solely in terms of attribute values. For example, the *Julia* and *John* proplets specify their functor as know, while the *know* proplet specifies Julia and John as its arguments. Because of their nonempty sur(face) slots, the proplets are language proplets, in contradistinction to the proplets in “[Concatenation by cross-copying](#)” and “[Coordination in big blue square](#)”, which are context proplets. For storage and retrieval, a proplet is specified uniquely¹² by its core and prn values (primary key). This suggests a two-dimensional database schema, as in a classic network database (Elmasri and Navathe 2010). However, instead of using member and owner records, DBS uses member proplets and owner values.

The result is called a word bank. Its database schema consists of a column of owner values in their alphabetical order (vertical). Each owner value is preceded by an empty slot, called the now front, and a list of member proplets (horizontal); together they constitute a *token line*.¹³

As an example, consider storing “[Content of Julia knows John. as a set of proplets](#)” as a nonlanguage content:

¹² Propositions containing two or more proplets with the same values, as in *Suzy loves Suzy*, require extra attention. They constitute a special case which (1) rarely occurs and (2) is disregarded here because it may be easily handled by the software.

¹³ The token line for any core value is found by using a trie structure (Briandais 1959; Fredkin 1960). The search for a proplet within a token line may use the prn value of the address in relation to the strictly linear increasing prn values. As pointed out by J. Handl (2008, 2012), this may be based on binary search, in time $O(\log(n))$ (Cormen et al. 2009), or interpolation, in time $O(\log(\log(n)))$ (Weiss 2005), where n is the length of the token line.

Storing “Content of Julia knows John, as a set of proplets” in a word bank

	member proplets	now front	owner values
...	[noun: John cat: snp fnc: know prn: 625 noun: Julia cat: snp fnc: know prn: 625 verb: know cat: #n-s3' #a' decl arg: Julia John prn: 625]		John
...			Julia
...			know

The proplets in a token line all have the same core value and are in the temporal order of arrival, reflected by their prn values (Hausser 2006, Sects. 11.2, 11.3).

In contrast to the task of designing a practical schema for arranging the books in a private library, the sorting of proplets into a word bank is simple and mechanical. The letter sequence of a proplet’s core value completely determines its token line for storage: the storage location for any new arrival is the penultimate position (now front) in the corresponding token line. When this slot is filled, the now front is reopened by moving the owner value one slot to the right (or, equivalently, pushing the member proplets one slot to the left, as in a push-down automaton).

By storing content like *sediment*, the stored data are never modified and any need for checking consistency is obviated. Changes of fact are written to the now front, like diary entries recording changes of temperature. Current data which refer to old ones use addresses as core values, implemented as pointers.

Cycle of natural language communication

The transfer mechanism of content from the speaker to the hearer is based on external surfaces which have neither a meaning nor any grammatical properties (“Physical framework of communication”). They must, however, belong to a language which the speaker and the hearer have each learned.

The learning enables the hearer to (1) recognize surfaces, (2) use the recognized but otherwise unanalyzed surfaces for looking up lexical entries which provide the meaning and the grammatical properties, and (3) connect them with the semantic relations of functor-argument and coordination. The learning enables the speaker to (1) navigate along the semantic relations between proplets, (2) produce language-dependent word form surfaces from the core values of proplets traversed, and (3) handle function word¹⁴ precipitation, micro word order, and agreement.

Successful communication between a speaker and a hearer is defined as follows:

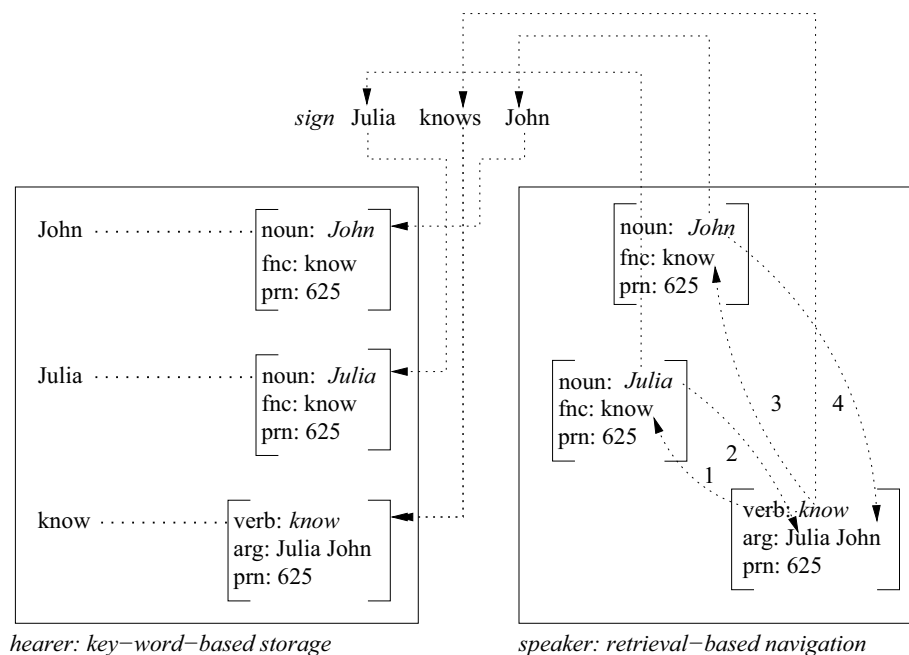
¹⁴ Examples of function words in English are determiners like a(n), the, some, every, all, prepositions like in, on, above, below, auxiliaries like be, have, do, coordinating conjunctions like and, or, and subordinating conjunctions like that, who, which, when, because.

Definition of successful communication

Natural language communication is successful if the content, mapped by the speaker into a sequence of external word form surfaces, is reconstructed and stored equivalently by the hearer.

Using the word bank content of “**Julia knows John**,” the transfer of information from the speaker to the hearer, based solely (1) on unanalyzed external surfaces and (2) the data structure and database schema of the agents’ cognition, may be shown schematically as follows:

Natural language transfer mechanism



The speaker’s navigation through a set of connected proplets serves as the conceptualization (*what to say*) and as the basic serialization (*how to say it*) of natural language production (McKeown 1985; Kass and Finin 1988). The hearer’s interpretation consists in deriving a corresponding set of proplets, based on automatic word form recognition and syntactic-semantic parsing. The time-linear order of the sign induced by the speaker’s navigation is eliminated in the hear mode, allowing storage of the proplets in accordance with the database schema of the content-addressable¹⁵ word bank. When the agent switches into the speak mode, order is reintroduced by navigating along the semantic relations between the proplets.

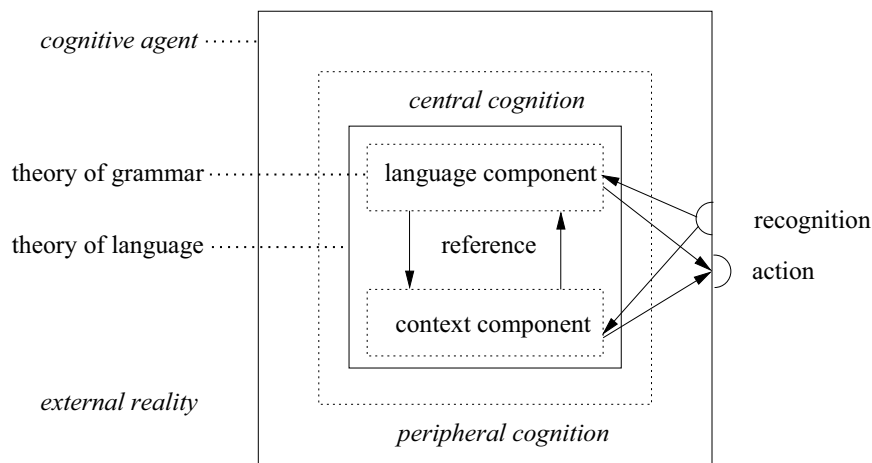
¹⁵ As a database, a word bank is content-addressable because it does not use an index (inverted file), in contrast to the widely used coordinate-addressable databases (RDBMS). See Chisvin and Duckworth (1992) for an overview.

Conceptual reconstruction of reference

In DBS, a cognitive content is defined as a set of proplets connected by address. Proplets with a non-empty sur(face) slot (“[Content of Julia knows John. as a set of proplets](#)”) represent a language content. Proplets with an empty sur slot (“[Concatenation by cross-copying](#)”) represent a context content. Otherwise, language and context proplets are alike. This holds specifically for their storage and retrieval in a word bank, which is based solely on their core value and order of arrival.

Conceptually, however, the schema “[Basic structure of cognition](#)” may be refined by (1) separating the levels of language and context, (2) introducing the place of pragmatics as an interaction between the two levels, and (3) distinguishing peripheral and central cognition.

Conceptual view of interfaces and components



Externally, the agent’s interfaces for language and nonlanguage recognition are the same, as are those for language and nonlanguage action.¹⁶ Internally, however, raw input data are separated by peripheral cognition into language and nonlanguage content (diagonal input arrows). Conversely in action, which realizes a content as raw output data regardless of whether it originated at the language or at the context level (diagonal output arrows).

For example, as a sound pattern the surface blue square will have a meaningful interpretation at the language level by someone who has learned English, but be treated as an uninterpreted noise at the context level by someone who has not. Conversely, even though the action of denying entrance may be realized by telling to go away (originating at the language level) or by slamming the door (originating at the context level), both result in raw output data.

The distinction between the language and the context component provides a cognitive treatment of reference. Reference to an object in the agent’s current environment is called *immediate* reference, while reference to cognitive content existing only in the

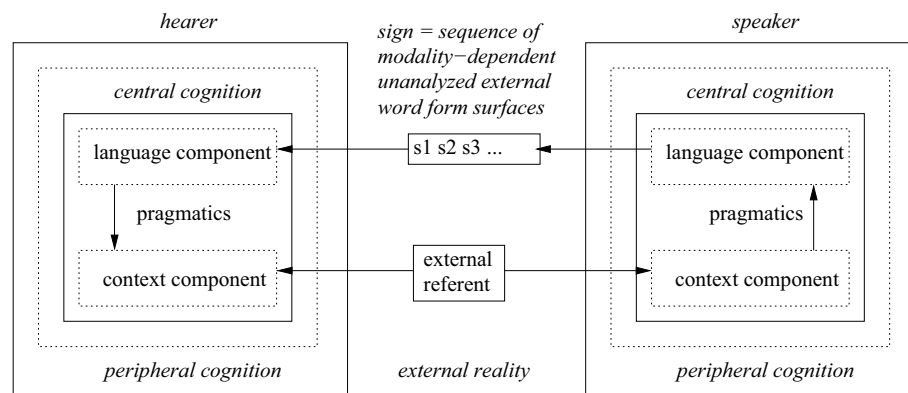
¹⁶ A differentiation into the sensory modalities (vision, audition, locomotion, manipulation) is omitted—not only for simplicity, but also because the meaning of a word or expression is independent of the modality of its external surface. For example, the meaning of the word square (“[Type and token of the concept square](#)”) is the same regardless of whether its surface is realized in speech, writing, or signing. A nonlanguage concept like the shape, color, taste, etc., of a blueberry may also be assumed to be independent from the modalities of its recognition (Hausser 2011, Sects. 2.2–2.4).

agents' memory, for example, J.S. Bach, is called *mediated* reference. For mediated reference, the agent-based ontology of DBS ([“Agent-based ontology”](#)) is essential.

As an example of immediate reference consider a speaker and a hearer in a common task environment (Newell and Simon 1972) and looking at a blue square. If the speaker says Take the blue square, the noun phrase refers to the object in question. Similarly for the hearer, for whom fulfilling the request requires reference to the same object.

Postulating an external relation between a surface and its referent would be a reification fallacy. Instead we reconstruct immediate reference cognitively:

Immediate reference as a purely cognitive procedure



Immediate reference relies on the agents' action and recognition interfaces for language (upper level) and the recognition of nonlanguage content (lower level). Mediated reference, in contrast, relies on language action and recognition (upper level) and the existence of corresponding content in the agent's memory. While immediate reference may be regarded as prototypical for the origin of language, it is a special case of mediated reference in that it has the additional requirement of context recognition (Hausser 2006, Sect. 2.5).

Terminological remark

Computer Science uses the term “reference” differently from analytic philosophy and linguistics. A computational reference is an address in a storage location. This may be coded as (1) a symbolic address (declarative) or as (2) a pointer to a physical storage location in the memory hardware (procedural). The term “generalized reference” is used in computational image reconstruction (computer vision).

In DBS, the term “reference” is used in the sense of philosophy and linguistics. However, the term is generalized insofar as no agent-external “representational token” is required ([“Constellations of generalized reference”](#), constellations 1 and 3).

Recanati (1997, 2004), Pelczar and Rainsbury (1998), and others use the term “generalized reference” for an analysis of the sign kind name which allows the surface Mary, for example, to refer to several different individuals. This is in contradistinction to Russell (1905) whose definite description¹⁷ analysis of “proper” names requires a unique referent.

¹⁷ For a modern variant of this approach see Burgess (2012).

The DBS analysis of names in “[Reference by baptism \(name\)](#)” also allows different referents (“[Name referring with multiple referents](#)”). However, while the “generalized reference” of Recanati, Pelczar et al., and others is based on assimilating names to the parameters of indexicals, the DBS analysis is traditional in that it is based on an act of baptism which is generalized in that it may occur implicitly as well as explicitly. Moreover, generalized reference in DBS is not limited to names, but includes reference by means of matching concepts (symbol) and pointing (indexical). In addition, DBS treats coreference (identity) by means of address, which occurs as a variant of matching and pointer reference, and provides object permanence (“[Object permanence by using address](#)”) for baptism-based reference.

Reference by matching (symbol)

The reference mechanism based on matching uses the type-token relation (“[Type and token of the color called blue](#)”, “[Type and token of the concept called square](#)”) and is associated with the sign kind *symbol*. For example, the terms a blue square and blue squares in the sentence sequence John saw a blue square. ... Blue squares are rare. are related as follows:

Reference with language proplets in token lines

	<i>member proplets</i>	<i>now front</i>	<i>owner values</i>
$\left[\begin{array}{l} \text{sur:} \\ \text{adj: blue (token)} \\ \text{mdd: square} \\ \text{prn: 41} \end{array} \right]$	$\dots \left[\begin{array}{l} \text{sur: blue} \\ \text{adj: blue (type)} \\ \text{mdd: square} \\ \text{prn: 48} \end{array} \right]$		blue
$\dots \left[\begin{array}{l} \text{sur:} \\ \text{noun: square (token)} \\ \text{sem: indef sg} \\ \text{fnc: see} \\ \text{mdr: blue} \\ \text{prn: 41} \end{array} \right]$	$\dots \left[\begin{array}{l} \text{sur: square} \\ \text{noun: square (type)} \\ \text{sem: indef pl} \\ \text{fnc: rare} \\ \text{mdr: blue} \\ \text{prn: 48} \end{array} \right]$		square

Stored in the agent’s word bank, the vertical relation between the language and the context component shown in “[Conceptual view of interfaces and components](#)” reappears as a horizontal relation between proplets within token lines. The proplets with the prn value 48 are the language proplets (non-empty sur slots), those with the prn value 41 are the context proplets (empty sur slots). Reference by matching holds between the two *blue* proplets with the prn values 41 and 48 and similarly between the two *square* proplets. The distinction between the type and the token, here indicated after the core values, is usually left implicit.

The combination of the proplets *blue* and *square* by means of a functor-argument relation is coded by the features [mdd: square] and [mdr: blue], respectively. The noun proplet with the feature [sem: indef sg] is an indefinite singular, that with the feature [sem: indef pl] is an indefinite plural.

Next consider the same reference relation without language. The missing sur values are emphasized with “Ø.”

Reference by matching without language

	<i>member proplets</i>	<i>now front</i>	<i>owner values</i>
$\left[\begin{array}{l} \text{sur:} \\ \text{adj: blue (token)} \\ \text{mdd: square} \\ \text{prn: 41} \end{array} \right]$	$\dots \left[\begin{array}{l} \text{sur: } \emptyset \\ \text{adj: blue (type)} \\ \text{mdd: square} \\ \text{prn: 48} \end{array} \right]$		blue
$\dots \left[\begin{array}{l} \text{sur:} \\ \text{noun: square (token)} \\ \text{sem: indef sg} \\ \text{fnc: see} \\ \text{mdr: blue} \\ \text{prn: 41} \end{array} \right]$	$\dots \left[\begin{array}{l} \text{sur: } \emptyset \\ \text{noun: square (type)} \\ \text{sem: indef pl} \\ \text{fnc: rare} \\ \text{mdr: blue} \\ \text{prn: 48} \end{array} \right]$		square

Here the reference relation holds between two nonlanguage contents—and not between a language content (meaning₁) and a nonlanguage content, as in “[Reference with language proplets in token lines](#)”.

Even though the reference relation is established between two individual proplet pairs in the same token lines, the combination into the complex content corresponding to blue square is accommodated as well¹⁸: in order to match, the two *blue* proplets must not only have the same¹⁹ core value, but also the same mdd continuation value, here square, and correspondingly for the mdr values of the two *square* proplets. Their fnc and prn values, however, are different.

Generalizing reference by matching to include referring with nonlanguage content results in the following constellations:

Constellations of generalized reference

1. *Nonlanguage content referring to nonlanguage content* Example: Agent sees something and identifies it with something seen before.
2. *Language content referring to nonlanguage content* Example: Agent describes a landscape in speak mode.
3. *Nonlanguage content referring to language content* Example: Agent identifies a current nonlanguage recognition with something it has read (for example, in a guide book) or heard about before.
4. *Language content referring to language content* Example: Agent describes what it has heard or read.

Cognitive agents without language are capable of reference constellation 1 only, while agents with language may use all four.

Reference by pointing (indexical)

The second reference mechanism of cognition is based on pointing. In natural language, it is illustrated by the indexical signs, such as the pronouns. The first step toward a computational implementation is the linguistic observation that the indexicals point at only

¹⁸ Apparently, Aristotle struggled to reconcile reference with content combination (Modrak 2001).

¹⁹ Disregarding the type-token distinction.

five different parameters, namely (1) first person, (2) second person, (3) third person, (4) place, and (5) time.

In English, the pronouns I, me, mine, we, and us point at the parameter for first person. The pronoun you points at the parameter for second person. The pronouns he, him, his, she, her, it, they and them point at the parameter for third person. The indexical adjs here and there point at the parameter for place. The indexical adjs now, yesterday, and tomorrow point at the parameter for time.

The indexical nouns pointing at the parameters of first, second, and third person are varied by grammatical distinctions. Consider the following examples illustrating grammatical variation in 1st person pronouns of English:

1st person pronouns distinctions

[sur: I noun: pro1 cat: s1 sem: sg fnc: mdr: nc: pc: prn:	[sur: me noun: pro1 cat: obl sem: sg fnc: mdr: nc: pc: prn:	[sur: we noun: pro1 cat: p1 sem: pl fnc: mdr: nc: pc: prn:	[sur: us noun: pro1 cat: obl sem: pl mdr: fnc: nc: pc: prn:
---	---	--	---

The proplets all share the indexical pointer pro1 as their core value. The different cat values s1 (first person singular), p1 (first person plural), and obl (oblique) control verb agreement, preventing, for example, ungrammatical *Me saw a tree or *Peter saw we. Ungrammatical *I sees a tree and *he see a tree are prevented by using the different cat values s1 (singular 1st person) and s3 (singular 3rd person).

Noun proplets of the sign kind indexical combine in the same way into propositions as proplets of the sign kind symbol or name. Consider the DBS analysis of English I heard you.:

Representing I heard you. as a language content

[sur: I noun: pro1 cat: s1 sem: sg fnc: hear mdr: nc: pc: prn: 71	[sur: heard verb: hear cat: #n' #a' decl sem: past arg: pro1 pro2 mdr: nc: pc: prn: 71	[sur: you noun: pro2 cat: sp2 sem: fnc: hear mdr: nc: pc: prn: 71
---	--	---

The question raised by this example is how the indexical pointers pro1 and pro2 are to be interpreted pragmatically relative to a context of use.

This leads to the second step of modeling the indexical reference mechanism. It is based on combining a propositional content with a cluster of parameter values of the agent's current STAR (Hausser 1999, Sect. 5.3). The STAR is an acronym for (1) location (Space), (2) time (Time), (3) self-identity (Agent), and (4) intended addressee (Recipient).

The STAR has two functions: (a) keeping track of the agent's current situation (orientation) and (b) providing referents for indexicals occurring in contents.²⁰ A STAR is coded as a proplet with the value of the A attribute serving as the core value and as the owner. In a word bank, a temporal sequence of STARs records the output of the agent's on-board orientation system and is listed as a token line:

Token line example of STARs defined as proplets

<i>member proplets</i>				<i>now front</i>	<i>owner value</i>
[S: kitchen T: $t_1 - t_2$ A: Sylvester R: Speedy 3rd: prn: 63–70]	[S: kitchen T: $t_{2+1} - t_3$ A: Sylvester R: Speedy 3rd: prn: 71–78]	[S: living room T: $t_{3+1} - t_4$ A: Sylvester R: Tweety 3rd: Speedy prn: 79–82]	[S: garden T: $t_{4+1} - t_5$ A: Sylvester R: Hector 3rd: prn: 83–87]		Sylvester

In addition to attributes represented by the letters of the STAR, there is a fifth, called 3rd, for third person indexicals. Though not required for the agent's basic orientation, 3rd is needed to provide the referent for items which are neither 1st nor 2nd person.²¹ As indicated by the prn values, e.g. [prn: 63–70], several consecutive propositions may share the same STAR.²²

In natural language communication, three perspectives on content must be distinguished (Hausser 2011 Chaps. 10, 11). The STAR-0 is the agent's perspective onto its current environment; it need not involve language. The STAR-1 is the agent's speak mode perspective onto stored content as required for language production; if ongoing events are reported directly, the STAR-1 equals the STAR-0. The STAR-2 is the agent's hear mode perspective onto language content as needed for the correct interpretation of indexicals. As an example of a STAR-0 perspective, consider the non-language content corresponding to I hear you:

²⁰ Integrating the interpretation of indexicals into the agent's on-board STAR orientation may be seen as an enhancement of Montague (1973), who's sign-oriented approach uses arbitrary parameter values, i.e. $i \in I$ for a moment of time and $j \in J$ for a possible world (space, location), superscripted at the end of a lambda expression. Introducing additional parameters for 1st, 2nd, and 3rd person, as has been suggested, has been made light of by Cresswell's (1972, p. 4) joking proposal of a "next drink parameter." The parameter approach has resurfaced as "Variablism," i.e. the view that names and pronouns should be treated semantically as variables (Cumming 2008).

²¹ According to King (2014), a context consists of time, location, agent, and world. In the STAR of DBS, S corresponds to King's location, T to time, and A to agent. The DBS counterpart to King's world are R (intended recipient, you) and 3rd (everyone and everything that is neither A nor R). However, DBS distinguishes between the STAR parameters as the agent's on-board orientation system and basis for interpreting indexicals, on the one hand, and the context as a selectively activated content in memory, on the other.

²² Instead of the names John, Mary, etc., usually employed in linguistic examples, let us use the animation characters of Sylvester the cat, Speedy Gonzales the mouse, Tweety the bird, and Hector the dog, familiar from TV, as an aid to distinguish the individuals pointed at by indexicals.

Anchoring a content to a STAR-0

[sur: noun: pro1 cat: s1 sem: sg fnc: hear mdr: nc: pc: prn: 63	[sur: verb: hear cat: #n' #a' decl sem: pres arg: pro1 pro2 mdr: nc: pc: prn: 63	[sur: noun: pro2 cat: sp2 sem: fnc: hear mdr: nc: pc: prn: 63	[S: kitchen T: 2013-09... A: Sylvester R: Speedy 3rd: prn: 63
---	--	---	--

This content differs from representing “I heard you.” as a language content because (1) it is nonlanguage (no sur values), (2) the sem value of the verb is pres (present tense) rather than past, and (3) a STAR is attached by having the same prn value as the content, here 63.

The STAR-0 shows the perspective of the agent Sylvester on his current environment. The S value specifies the location as the kitchen, the pres value of the verb points at the T value, the indexical pro1 points at the A value Sylvester the cat, and pro2 points at the R value Speedy the mouse.

Next Sylvester realizes the content in language by saying to Speedy I heard you. As time has moved, the language content representing “I heard you” is anchored to a second, later STAR-0 with the prn value 71. From these two STAR-0, the agent computes the following STAR-1 perspective for the language content “Representing I heard you. as a language content”:

Speak mode anchoring to a STAR-1

[sur: I noun: pro1 cat: s1 sem: sg fnc: hear mdr: nc: pc: prn: 64	[sur: heard verb: hear cat: #n' #a' decl sem: past arg: pro1 pro2 mdr: nc: pc: prn: 64	[sur: you noun: pro2 cat: sp2 sem: fnc: hear mdr: nc: pc: prn: 64	[S: kitchen T: 2013-09... A: Sylvester R: Speedy 3rd: prn: 64
---	--	---	--

The agent’s perspective is looking from his present situation back on the stored content “Anchoring a content to a STAR-0” and encoding it in language. The content is connected to the agent’s current STAR via the common prn value 64. The content differs from that of “Anchoring a content to a STAR-0” in (i) the sem value past (rather than pres) of the verb proplet and (ii) the language-dependent sur values of the content proplets.

When the language content “I heard you” is interpreted by the addressee (recipient), Speedy the mouse uses the content of the language sign and its current STAR-0 to derive the STAR-2 perspective. The result is as follows:

STAR-2 perspective in hear mode

[sur: noun: pro2 cat: sp2 sem: fnc: hear mdr: nc: pc: prn: 53]	[sur: verb: hear cat: #n' #a' decl sem: past arg: pro2 pro1 mdr: nc: pc: prn: 53]	[sur: noun: pro1 cat: obl sem: sg fnc: hear mdr: nc: pc: prn: 53]	[S: kitchen T: 2013-09... A: Speedy R: Sylvester 3rd: prn: 53]
--	---	---	---

Speedy as the interpreting agent uses his personal prn value and a different STAR: compared to the STAR of Sylvester, the A and R values are reversed and Sylvester's I heard you is reinterpreted by Speedy's STAR-2 perspective as you heard me.

Case study

The use of pronouns in indexical use has been illustrated with a case study which analyzes a dialog (Hausser 2011, Chapter 10). It illustrates the production and interpretation of indexicals with explicit STARs for the speak and the hear mode in statements, WH and Yes/No questions with answers, and a request with fulfillment. The use of pronouns in coreferential use is systematically investigated in Chapter 11 of Hausser (2011).

Reference by baptism (name)

Just like the other two reference mechanisms, the baptism-based reference of names is implemented as a cognitive operation. It consists of (1) establishing *object permanence*²³ and (2) *baptism* based on cross-copying between a name and its referent.

For implementing object permanence, DBS uses identity by address. For example, when a robot observes an unfamiliar dog running through the bushes, it must understand that the different appearances are instances of the same referent. This recognition interpretation, which is based on yet another type-token relation, is coded by using an address as the core value of the non-initial proplets, pointing at the proplet representing the initial appearance of the referent:

Object permanence by using address

<i>member proplets</i>			<i>now front owner value</i>	
[sur: noun: dog ... prn: 83]	[sur: noun: (dog 83) ... prn: 99]	[sur: noun: (dog 83) ... prn: 112]	[sur: noun: (dog 83) ... prn: 131]	dog

The different prn values indicate that each member proplet is part of a different proposition, allowing different continuation values. The core values (dog 83) of the non-initial member proplets point at the initial proplet, which is the referent and formally recognizable by its *non-address*²⁴ core value.

²³ The notion of object-permanence originated in cognitive psychology (Piaget 1954). It may be regarded as a non-truth-conditional, non-modal counterpart to "rigid designators" (Kripke 1972).

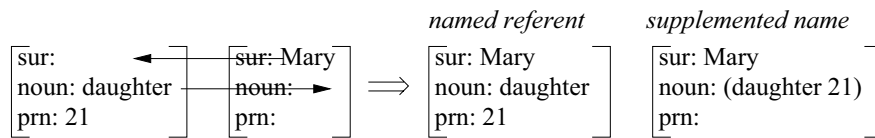
²⁴ While continuation values are always addresses, core values may be non-address or address.

A token line like “[Object permanence by using address](#)” may contain several initial dog referents, each referring to another individual. They are distinguished by their different prn values and the address numbers of the associated coreferent proplets. This is sufficient for the agent to properly discriminate between different dog referents in cognition and between their sets of coreferent proplets, all in the same token line.²⁵

It is not sufficient, however, for language communication. This is because the prn values of referents are not synchronized between agents: agents may encounter the same referent at different occasions, and the number of coreferent items may differ between agents. What is needed is a name surface and an interagent consensus on which item(s) the name refers to. The consensus is simply achieved: the not yet initiated agent follows the practice observed because communication would break down otherwise (no private language!).

The DBS implementation is based on (1) a lexical name proplet which has a sur(face) value but no core value and (2) a connected referent proplet which has a core value but no sur value. The two proplets are supplemented by an event of generalized baptism which cross-copies the sur value of the name into the sur slot of the referent and the core value of the referent into core slot of the name.

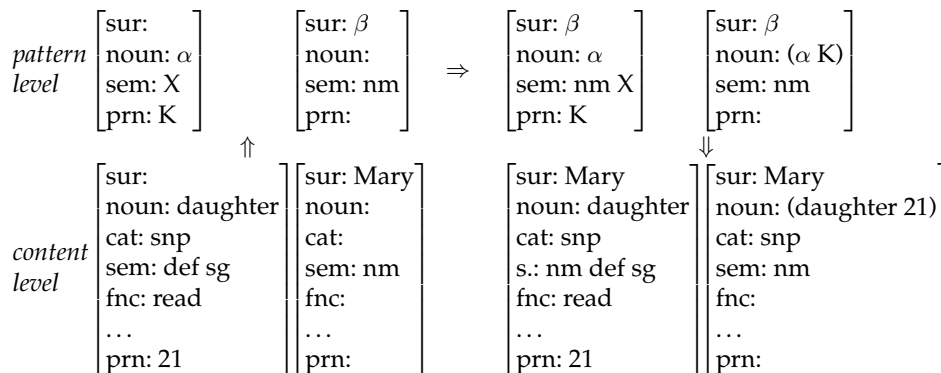
Baptism as cross-copying



The named referent proplet is stored in the token line of the core value and used in the speak mode. The supplemented name proplet is stored in the token line of the surface and used in the hear mode.

The baptizing event is formalized as the following DBS operation:

Applying the formal baptizing operation



The third proplet at the content level is the named referent. As a member proplet in the agent’s word bank, it has a prn value, here 21. The supplemented sur value is used in the speak mode to realize the name surface. The fourth proplet is the supplemented name.

²⁵ In agents without language, it is a pre-stage of naming (part of case 1 in “[Constellations of generalized reference](#)”).

Its core value is the address of the referent. The prn value will be provided by a hear mode derivation (e.g. Hausser 2006, Sect. 3.4.2).

The two proplets resulting in “[Applying the formal baptizing operation](#)” look similar, but they are stored in different token lines and used in the different roles of the speaker and the hearer. Consider the following word bank containing three referents named Mary, referring to the grandmother, the mother, and the daughter in a family. The token lines are in the alphabetical order *daughter*, *grandmother*, *Mary*, *mother*.²⁶

Name referring with multiple referents

member proplets		now front	owner values
	<div> <div> sur: Mary noun: daughter cat: snp sem: nm f fnc: sing ... prn: 21 </div> </div>		daughter
<div> <div> sur: Mary noun: grandm. cat: snp sem: nm f fnc: cook ... prn: 7 </div> </div>			grandmother
<div> <div> sur: Mary noun: (grandm. 7) cat: snp sem: nm f fnc: ... prn: </div> </div>	<div> <div> sur: Mary noun: (mother 14) cat: snp sem: nm f fnc: ... prn: </div> </div>	<div> <div> sur: Mary noun: (daughter 21) cat: snp sem: nm f fnc: ... prn: </div> </div>	Mary
	<div> <div> sur: Mary noun: mother cat: snp sem: nm f fnc: read ... prn: 14 </div> </div>		mother

The member proplets show the result of three baptism operations like “[Applying the formal baptizing operation](#)”. In the token line of *Mary*, each supplemented name proplet occurs only once. Supplemented names are not written into the lexicon because a core value like (daughter 21) is not a convention of the natural language at hand. Instead it is the result of a generalized baptism event witnessed by the agent. Nevertheless, the supplemented name proplets in the *Mary* token line have a lexical quality in that they have

²⁶ For clarity, each supplemented name proplet and its referent are shown in the same column. In reality, token lines order proplets contiguously, without any empty positions, solely in accordance with the proplets’ arrival. Thus, token lines are “dense” and independent of each other.

neither continuation nor prn values—like the lexical proplets resulting from automatic word form recognition.²⁷

When the hearer interprets a sentence containing a name, the name activates the corresponding token line, here that of Mary. The choice between different referents, here the grandmother, the mother, and the daughter, may have one of the following results: (1) the chosen referent equals the one intended by the speaker (correct choice), (2) does not equal the one intended by the speaker (incorrect choice), or (3) no referent is chosen (inconclusive result). The choice between multiple potential name referents is usually not at random, however. Instead, the referent most suitable to the utterance situation will usually be the correct one. If uncertainty remains, the hearer may ask the speaker for clarification.

For an agent in the speak mode, there is no ambiguity. Instead, the speaker selects the intended referent, e.g. (daughter 21). If the agent acquired the appropriate name in the hear mode (“[Applying the formal baptizing operation](#)”), it is preserved in the word bank (“[Known name refers to known referent](#)”) and may be used in the speak mode. If the agent is in the position to select and bestow a name, it is also available for realization.

Depending on whether or not an agent’s word bank already has a token line (1) for a certain name and (2) proplets for its referent(s), the following case distinctions may be described in the name interpretation of the hearer:

Case distinctions of name reference

1. the name and the referent have been used before,
2. the name has been used before, but the referent is new,
3. the name has not been used before, but the referent exists, or
4. neither the name nor the referent have been used before by the agent.

As an example of case 1, consider agent A observing daughter Mary eating a cookie. Later agent A reports to agent B Mary ate a cookie. Assuming that agent B knows the intended referent of Mary, the following proposition would be stored in the word bank of agent B (hear mode):

Known name refers to known referent

$\left[\begin{array}{l} \text{sur: mary} \\ \text{noun: (daughter 21)} \\ \text{fnc: eat} \\ \text{prn: 102} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{verb: eat} \\ \text{arg: (daughter 21) cookie} \\ \text{prn: 102} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{noun: cookie} \\ \text{fnc: eat} \\ \text{prn: 102} \end{array} \right]$
--	--	--

The proplets are held together by the common prn value 102. As a named referent, the first proplet is stored in the token line of the core value. The sur values are not preserved

²⁷ Our computational reconstruction of the baptism event based on cross-copying differs from other theories of naming such as the descriptive theory of proper names (Russell 1905) and the rigid designator analysis (Kripke 1972), which use the perspective of an external observer instead of an agent-based ontology. The formal DBS treatment of baptism-based reference (“[Baptism as cross-copying](#)”) provides a simple, efficient procedural implementation suitable for building a talking robot.

in storage.²⁸ Instead they will be provided by language-dependent lexicalization rules in the speak mode. The address values point at the initial referent (object permanence, “Object permanence by using address”).

The second case is illustrated by agent B knowing the grandmother and mother referents of Mary, but not the daughter referent. This means that the token line of *Mary* exists in the word bank of agent B, though without a supplemented name proplet pointing at the referent proplet *daughter*. The token line of *daughter* may not even contain a content proplet suitable as the referent. The two missing proplets may be added to agent’s B word bank by agent A saying Did you know that Mary’s daughter is called Mary as well?

The third case is illustrated by an unfamiliar dog running through the bushes. Its proplet representation is added to the existing token line for dog. This still unnamed initial referent is recognizable by its non-address core value. When the dog is called Fido, the name is written into the sur slot of initial dog proplet (named referent, “Applying the formal baptizing operation”) and the supplemented name proplet is stored in the *Fido* token line. If the name has not been used before, a new token line is created for it by defining a token line using the name’s surface as the owner value.

The fourth case is illustrated by meeting an unknown animal in a zoo and learning that it is called Rosie. Thus a word bank must provide for two dimensions of continuous extension: (1) the time-linear lengthening of token lines (horizontal) and (2) the insertion of new token lines into the column of owner values (vertical).

Reference by address (coreference)

Coreference by address occurs with all three sign kinds. In reference by baptism (name) it is the only mechanism for relating the supplemented name to the named referent. In reference by matching (symbol) and by pointing (indexical), in contrast, it is an additional mechanism. Let us begin with the coreferential interpretation of a symbol in the sequence a unicorn ... the unicorn:

Coreference in a token line

<i>member proplets</i>		<i>now front</i>	<i>owner value</i>
$\left[\begin{array}{l} \text{noun: unicorn} \\ \text{sem: indef sg} \\ \text{prn: 23} \end{array} \right]$...	$\left[\begin{array}{l} \text{noun: (unicorn 23)} \\ \text{sem: def sg} \\ \text{prn: 26} \end{array} \right]$	unicorn

The coreference relation is coded by using the address of the first proplet as the core value of the second. The second proplet must be a definite noun.

The other sign kind with a coreferential interpretation in addition to its own reference mechanism is the indexical. The coreferential interpretation of a pointer (e.g. 3rd person pronoun) uses the address of a preceding (antecedent) or following (postcedent) referent, instead of referring to the STAR. Reusing the example in “Reference by pointing (indexical)”, consider the extrapositional coordination “Speedy hid in the cupboard. I heard him.” uttered by Sylvester and interpreted by Hector.

²⁸ The name proplet preserves the sur value resulting from the baptism operation as a marker, written in default font and in lower case. The marker is used for surface realization in the speak mode.

The question is whether *him* refers indexically, e.g. to Tweety the bird, or coreferentially to Speedy the mouse. For the speaker Sylvester the answer is clear because language expressions are not ambiguous in the speak mode. For Hector as the hearer, however, the answer is unknown. For successful communication, Hector must use the same reference mechanism, indexical or coreferential, as Sylvester.

Formally, the two possible hear mode interpretations compare as follows:

Indexical interpretation of *him*

$$\begin{array}{c} \left[\begin{array}{l} \text{noun: Speedy} \\ \text{fnc: hide} \\ \text{prn: 78} \end{array} \right] \left[\begin{array}{l} \text{verb: hide} \\ \text{arg: Speedy} \\ \text{mdr: cupboard} \\ \text{nc: (hear 79)} \\ \text{prn: 78} \end{array} \right] \left[\begin{array}{l} \text{noun: cupboard} \\ \text{mdd: hide} \\ \text{prn: 78} \end{array} \right] \\ \\ \left[\begin{array}{l} \text{noun: pro2} \\ \text{fnc: hear} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{verb: hear} \\ \text{arg: pro2 pro3} \\ \text{pc: (hide 78)} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{noun: pro3} \\ \text{fnc: hear} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{S: garden} \\ \text{T: 2013-09-23T08:20:00} \\ \text{A: Hector} \\ \text{R: Sylvester} \\ \text{3rd: Tweety} \\ \text{prn: 78-79} \end{array} \right] \end{array}$$

This hear mode interpretation uses Hector's STAR with a STAR-2 conversion²⁹ of the content from *pro1* to *pro2*. Represented as *pro3*, *him* points at the Tweety value of the 3rd attribute in the STAR.³⁰

Next consider Hector's coreferential interpretation of *him* as referring to Speedy:

Coreferential interpretation of *him*

$$\begin{array}{c} \left[\begin{array}{l} \text{noun: Speedy} \\ \text{fnc: hide} \\ \text{prn: 78} \end{array} \right] \left[\begin{array}{l} \text{verb: hide} \\ \text{arg: Speedy} \\ \text{mdr: cupboard} \\ \text{nc: (hear 79)} \\ \text{prn: 78} \end{array} \right] \left[\begin{array}{l} \text{noun: cupboard} \\ \text{mdd: hide} \\ \text{prn: 78} \end{array} \right] \\ \\ \left[\begin{array}{l} \text{noun: pro2} \\ \text{fnc: hear} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{verb: hear} \\ \text{arg: pro2 (Speedy 78)} \\ \text{pc: (hide 78)} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{noun: (Speedy 78)} \\ \text{fnc: hear} \\ \text{prn: 79} \end{array} \right] \left[\begin{array}{l} \text{S: garden} \\ \text{T: 2013-09-23T08:20:00} \\ \text{A: Hector} \\ \text{R: Sylvester} \\ \text{3rd:} \\ \text{prn: 78-79} \end{array} \right] \end{array}$$

The representation of the first proposition is the same as in “[Indexical interpretation of *him*](#)”. The STAR is different, however, in that its 3rd attribute has no value. Represented as the address value (Speedy 78), *him* is interpreted coreferentially.

²⁹ Hausser (2011), Chap. 10.

³⁰ The indexical interpretation of *pro2* and *pro3* is defined to refer to the R and 3rd values, respectively, of the STAR. For a more extensive treatment see Hausser (2011) Chaps. 10, 11.

De dicto/de re

A puzzle in ancient philosophy of language has been called the *de dicto/de re* ambiguity by medieval logicians, e.g. Thomas Aquinas (1225–1274) and William of Occam (1288–1347).³¹ More recently, roughly the same distinction has been called *uneven/even* by Frege (1892), *opaque/transparent* by Quine (1960), and *intensional/extensional* by Montague (1973). The *de dicto/de re* ambiguity occurs with certain verbs and modal operators.

The *de dicto* reading arising with verbs constitutes an exception to two fundamental laws of medieval semantics, known as (1) Existential Generalization and (2) Substitutivity of Identicals (Hausser 1999, Sect. 20.4). Existential Generalization fails if the meaning of a noun is equated with its real world referent (ontological assumption) and the referent doesn't exist.

This is illustrated by Montague (op. cit.) as follows:

Different entailments of existence

a. *De re*

John **finds** a unicorn. > There exists a unicorn.

b. *De dicto/de re*

John **seeks** a unicorn. $\not>$ There exists a unicorn.

In example b, the entailment fails ($\not>$) if John is looking for any old unicorn (*de dicto*), because there might not exist one. It is also possible, however, that John is looking for a particular unicorn, in which case the referent exists (*de re*). Example a, in contrast, has only a *de re* reading (>).

This difference between examples a and b is caused by their different verbs. In the terminology of Montague, *find* creates an extensional context in which truth of the premise entails truth of the conclusion, while *seek* creates an intensional context in which truth of the premise does not. Technically, Montague treats the *de dicto/de re* alternative produced by an intensional verb as a scope ambiguity based on the position of the quantifier used to represent a unicorn in predicate calculus.

DBS takes a different approach: (1) reference is treated as a purely cognitive relation between two contents ("[Constellations of generalized reference](#)"), (2) there are neither quantifiers nor any of the scope ambiguities they create (Hausser 2006, Sect. 6.4), and (3) ambiguity arises only in recognition, including the hear mode,³² but not in action.

For example, if the agent is looking for a spoon (action) in an unfamiliar house, this content is represented as follows:

³¹ For an overview see Dutilh Novaes (2004).

³² Even an ambiguity deliberately created in the speak mode ("diplomatic ambiguity," Pehar 2001) arises only for the hearer.

Nonlanguage action using de dicto reference

<i>member proplets</i>	<i>now front</i>	<i>owner value</i>
...	$\left[\begin{array}{l} \text{noun: spoon } type \\ \text{fnc: seek} \\ \text{prn: 67} \end{array} \right]$	spoon

The search would be successful if the context were to provide a token matching the type (“[Reference by matching without language](#)”). However, because the fnc value is the intensional verb *seek*, no assumption about the outcome is made at this point.

In a familiar house, in contrast, the agent may be looking for a particular spoon. In DBS, this de re constellation is represented as the following token line:

Nonlanguage action using de re reference

<i>member proplets</i>	<i>now front</i>	<i>owner value</i>
... $\left[\begin{array}{l} \text{noun: spoon } token \\ \text{fnc: have} \\ \text{prn: 67} \end{array} \right]$...	$\left[\begin{array}{l} \text{noun: (spoon 67)} \\ \text{fnc: seek} \\ \text{prn: 93} \end{array} \right]$	spoon

The second *spoon* proplet, here with the prn value 93, has an antecedent serving as the initial referent, here with the prn value 67. The agent refers by address to the spoon being searched for (“[Reference by address \(coreference\)](#)”).

If the action is language production, e.g. uttering “I am looking for a spoon”, the speaker knows full well whether the intended reference mechanism is “[Nonlanguage action using de dicto reference](#)” or “[Nonlanguage action using de re reference](#)”. In the hear mode, in contrast, there is a pragmatic ambiguity. Communication is successful if the hearer selects the reading meant by the speaker.

In DBS, the failing Existential Generalization on the hearer’s de dicto interpretation (“[Different entailments of existence](#)”) is implemented as an inference. The antecedent pattern ensures that the input noun proplet (1) has a non-address core value, (2) is indefinite, and (3) has an intensional verb as its fnc value. The consequent pattern derives the new proposition *the unicorn might not exist*.

The valid Existential Generalization of the corresponding de re interpretation is also derived by a hear mode inference. The antecedent pattern ensures that there is an initial referent which the *unicorn* proplet is coreferent with (as in “[Nonlanguage action using de re reference](#)”). The consequent pattern derives the new proposition *the unicorn exists* (no negation in the sem slot of the verb).

The de dicto/de re distinction in its application to a transitive verb construction may be summarized as follows. If the verb is extensional or the object noun is definite, the interpretation is de re for speaker and hearer. If the verb is intensional (in the sense of Montague) and the noun is indefinite, in contrast, the interpretation depends on whether or not the indefinite noun has an antecedent, i.e. an initial referent which it refers to by address (coreference). If the answer is yes (“[Nonlanguage action using de re reference](#)”), the interpretation is de re, otherwise de dicto (“[Nonlanguage action using de](#)”).

[dicto reference](#)"). For the speaker, who knows the answer, the intended interpretation is a matter of choice. For the hearer, who doesn't know the answer, the alternative interpretations constitute a pragmatic ambiguity.

Finally consider the nature of existence, the entailment of which is at the center of the de dicto/de re distinction ("[Different entailments of existence](#)"). In the ontology of an agent-based approach ("[Conceptual reconstruction of reference](#)"), there are two kinds: (1) real existence, as in immediate reference ("[Immediate reference as a purely cognitive procedure](#)"), and (2) assumed existence, as in mediated reference. For example, when the noun unicorn is used de re in a piece of fiction (mediated reference), the unicorn exists in the story, but the kind of existence is fictional in that no agent-external counterpart is entailed.

Conclusion

The construction of an artificial cognitive agent with language (talking robot) is an interdisciplinary project which requires the cooperation of linguistics, philosophy, psychology, artificial intelligence, cybernetics, mathematical complexity theory, and robotics. It challenges the participating sciences to cooperate within a consistent, comprehensive, computationally verifiable framework.

The linguistic part of designing a talking robot is reconstructing the cycle of natural language communication, consisting of the hear, the think, and the speak mode ("[Cycle of natural language communication](#)"). The philosophical part includes the computational reconstruction of the different mechanisms of generalized reference, as shown here in "Reference by matching (symbol)" to "[Reference by address \(coreference\)](#)". The part of cybernetics and artificial intelligence is the design of an autonomous control which is capable of dealing with unfamiliar situations in the agent's ecological niche, with long-term survival as the standard of success.

These are theoretical tasks which require the design of a data structure, a database schema, an algorithm, and a functional flow connecting the agent's recognition and action. They are essential for the hardware development of an artificial agent because they provide the robotic interfaces for recognition and action ("[Physical framework of communication](#)") with an agent-internal cognitive system to map into and out of.

A continuous cycle of upscaling and computational verification constitutes a self-correcting research strategy. Based on the identification and permanent correction of errors as well as the inclusion of more and more phenomena, systematic upscaling allows to approximate completeness of function and of data coverage

Direct access to the cognition of a talking robot via the service channel (Hausser 2006, Sect. 2.4.3) allows to view the reasoning of the artificial agent objectively, providing epistemology in philosophy and behavior modeling in psychology with the possibility of computational testing as a modern methodology. Regular software releases containing the improvements of the last upscaling cycle may be used by paying subscribers to satisfy their natural language processing needs, resulting in solid long-term funding.

Acknowledgements

The author would like to thank the anonymous reviewer for constructive comments, which helped to improve the manuscript.

Competing interests

The author declares that he has no competing interests.

Received: 29 January 2016 Accepted: 24 December 2016

Published online: 21 January 2017

References

- Barnes J (ed) (1974) The complete works of Aristotle. Princeton University Press, Princeton
- Barsalou L (2008) Grounded cognition. *Annu Rev Psychol* 59:617–645
- Biederman I (1987) Recognition-by-components: a theory of human image understanding. *Psychol Rev* 94:115–147
- de la Briandais R (1959) File searching using variable length keys. *Proc West Jt Comput Conf* 15:295–298
- Burgess A (2012) Metalinguistic descriptivism for Millians. *Australas J Philos* 91:443–457
- Chen F (2006) Human factors in speech interface design. In: Chen F (ed) *Designing human interface in speech technology*. Springer, Berlin, pp 167–224
- Chisvin L, Duckworth RJ (1992) Content-addressable and associative memory. In: Yovits MC (ed) *Advances in computer science*, 2nd edn. Academic Press, Cambridge, pp 159–235
- Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to algorithms*, 3rd edn. MIT Press, Cambridge
- Cresswell M (1972) The world is everything that is the case. *Australas J Philos* 50:1–13
- Cumming S (2008) Variabilism. *Philoso Rev* 117(4):525–554
- Dutilh Novaes C (2004) A medieval reformulation of the de dicto/de re distinction. In: *LOGICA yearbook 2003*. Prague, Filosofia, pp 111–124
- Elmasri R, Navathe SB (2010) *Fundamentals of database systems*, 6th edn. Benjamin-Cummings, Redwood City
- Fredkin E (1960) Trie memory. *Commun ACM* 3(9):490–499
- Frege G (1892) Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*. 100:25–50
- Handl J (2008) Entwurf und Implementierung einer abstrakten Maschine für die oberflächenkompositionale inkrementelle Analyse natürlicher Sprache. In: *Diplom thesis, Department of Computer Science, Univ, Erlangen Nürnberg*
- Handl J (2012) Inkrementelle Oberflächenkompositionale Analyse und Generierung Natürlicher Sprache. In: *Inaugural Dissertation, CLUE, Univ. Erlangen Nürnberg*. <http://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/3933>
- Hausser R (1989) Computation of language: an essay on syntax, semantics, and pragmatics in natural man-machine communication. *Symbolic computation: artificial intelligence*. Soft cover reprint 2013. Springer
- Hausser R (1992) Complexity in left-associative grammar. *Theor Comput Sci* 106(2):283–308
- Hausser R (1999) Foundations of computational linguistics. In: Hausser R (ed) *Human–computer communication in natural language*, 3rd edn. Springer, Berlin
- Hausser R (2005) Memory-based pattern completion in database semantics. *Lang Inf* 9(1):69–92
- Hausser R (2006) A computational model of natural language communication: interpretation, inference, and production in database semantics. Springer, Berlin
- Hausser R (2011) *Computational linguistics and talking robots: processing content in database semantics*. Springer, Berlin
- Kass R, Finin T (1988) Modeling the user in natural language systems. *Comput Linguist* 14:5–22
- King J (2014) Speaker intentions in context. *Noûs* 48(2):219–237
- Kripke S (1972) Naming and necessity. In: Davidson D, Harman G (eds) *Semantics of natural language*. D. Reidel, Dordrecht, pp 253–355
- McKeown K (1985) Discourse strategies for generating natural-language text. *Artif Intell* 27:1–41
- Modrak D (2001) *Aristotle's theory of language and meaning*. Cambridge Univ. Press, New York
- Montague R (1973) The proper treatment of quantification in ordinary English. In: Hintikka J, Moravcsik J, Suppes P (eds) *Approaches to natural language*. D. Reidel, Dordrecht, pp 221–242
- Newell A, Simon HA (1972) *Hum Probl Solving*. Prentice-Hall, Englewood Cliffs
- Peirce CS (1931–1935) *Collected papers*. In: Hartshorne C, Weiss P (ed) *Harvard Univ. Press, Cambridge*
- Pehar D (2001) *Language and diplomacy*. Lambert Academic Publishings, Saarbrücken
- Pelczar M, Rainsbury J (1998) The indexical character of names. *Synthese* 114:293–317
- Piaget J (1954) *The construction of reality in the child*. Basic Books, New York
- Quine WVO (1960) *Word and object*. MIT Press, Cambridge
- Recanati F (1997) *Direct reference: from language to thought*. Blackwell, Oxford
- Recanati F (2004) *Literal meaning*. Cambridge Univ. Press, Cambridge
- Reimer M, Michaelson E (2014) Reference. In: Zalta EN (ed) *The stanford encyclopedia of philosophy: winter 2014 edition*. <http://plato.stanford.edu/entries/reference/>
- Russell B (1905) On denoting. *Mind* 14:479–493
- Steels L (1999) The talking heads experiment, limited pre-edition for the Laboratorium exhibition. Antwerp, Belgium
- Weiss MA (2005) *Data structures and problem solving using Java*, 3rd edn. Pearson Addison-Wesley, Upper Saddle River
- Wiener N (1961) *Cybernetics, or control and communication in animal and machine*, 2nd edn. MIT press, Cambridge