Agent-Based Memory as an On-Board Database

Roland Hausser
Universität Erlangen-Nürnberg (em.)

Abstract

The databases of today are used for storage in, and retrieval from, large collections of systematic information, such as medical, insurance, and tax data. In natural agents, cognitive content is stored in, and retrieved from, the brains' memory. In artificial DBS agents, natural memory is reconstructed as an on-board database.

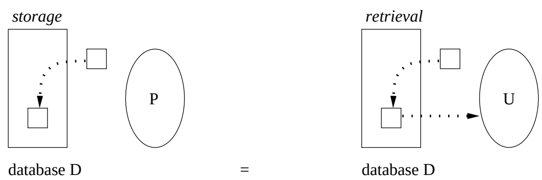
An agent-based database serving as the memory of an artificial agent requires an interface component for automatic recognition and action, and a data-driven application of operations. Basic tasks are (i) transfer of content from a speaker to a hearer in natural language communication (Sects. 1–5) (ii) coactivation of content stored in memory by current processing (Sects. 6–8), and (iii) an algorithm for efficient non-language recognition, for example in vision (Sect. 9), and action. The goal is *functional equivalence* between the natural prototype and its computational reconstruction at appropriate levels of abstraction.

keywords: raw data for content-transfer, speak mode, hear mode, now front

1 Input-Output of Conventional Database vs. On-Board Memory

A conventional database (databank) has the (i) input constellation of a programmer storing the data and the (ii) output constellation of a user retrieving copies of the data:

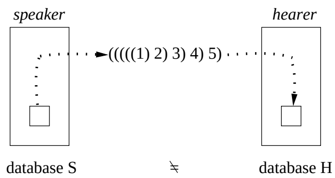
1.1 CONVENTIONAL DATABASE INTERACTION



Interaction takes place between two different agents, i.e. the programmer (oval P) and the user (oval U), and a single database, i.e. the large box D. The small boxes represent the content serving as input and output. The programmer P controls the storage and the user U controls the retrieval operations. Both use the same programming language, e.g. SQL, the commands of which are executed as electronic procedures.

Next consider the transfer of content from speaker to hearer by means of raw data:

1.2 SPEAKER AND HEARER INTERACTING IN COMMUNICATION



The large boxes represent the on-board memories of two agents, the speaker S and the hearer H. which may be natural or artificial, and typically contain different contents. In natural language communication, agents alternate between the speak and the hear mode (turn-taking). In the speak mode, automatic word form production takes cognitive content as input and maps it into language-dependent surfaces as agent-external raw data output. In the hear mode, automatic word form recognition takes agent-external raw data as input and maps it into cognitive content as output.

The raw data transporting content from the speaker to the hearer have the left-associative (time-linear) structure (Aho and Ullman 1977, p. 47) shown abstractly in 1.2 as (((((1) 2) 3) 4) 5). Otherwise the agent-external raw surface data have neither meaning nor any grammatical properties whatsoever (no reification in DBS), but may be measured by the natural sciences. For word form recognition and production, DBS uses computational pattern matching based on the type-token relation from philosophy (Peirce 1903).

2 Data Structure and Operations in a Record-Based Database

A computational database is defined by (i) a data structure, (ii) a database schema, and (iii) an algorithm for the storage, retrieval, and processing of content. In business, the most widely used data structure is *records*, as in the following example:

2.1 INFORMATION IN A RECORD-BASED DATABASE

	last name	**first name**	**place**	**...**
A1	Schmidt	Peter	Bamberg	...
A2	Meyer	Susanne	Nürnberg	...
A3	Sanders	Reinhard	Schwabach	...
	⋮	⋮	⋮	

The columns, named by different attributes like first name, last name, etc., are called the fields of the record type. The lines A1, A2, etc., each constitute a record.

The data structure determines the operations of the database. If the data structure is records, the standard operations are for the retrieval, update, and recombining of information. For example, to retrieve the name of the representative in Schwabach, the user types the following command in the programming language SQL (for "structured query language"):

2.2 QUERY OF THE DATABASE 2.2

Query:

```
select A#
where city = 'Schwabach'
```

Result:

```
result: A3 Sanders Reinhard
```

Other SQL commands are outer join and inner join for the conjunction or intersection of record columns. Change of data is carried out by replacement operations. Illegal modification such as theft and fraud is prevented by differentiating access privileges.

In summary, agent-driven (human) commands initiate quasi-mechanical procedures which correspond to storing, retrieving, re-sorting, and correcting cards in a filing cabinet. Compared to the nonelectronic method, the computational system has many practical advantages. The adding, finding, recombining, and correcting information is largely automated, making it faster, and the possibilities of search are more powerful because the records of different fields may be set-theoretically manipulated for complex queries.

3 Data Structure and Operations in DBS

Alternative to defining content in terms of records, DBS uses natural language grammar. The data structure is non-recursive feature structures with ordered attributes called *proplets*, connected into content by the four semantic relations of subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct, coded by address.

Consider the following representation of A3 (2.1) as a set of proplets:

3.1 THE CONTENT OF "R.S. resides in S." IN DBS

sur: reinhard sanders	sur:	sur: schwabach
noun: **[person x]**	verb: **reside**	noun: **[town y]**
cat: snp	cat: #n′ #mdr′ decl	cat: snp
sem: nm m sg	sem:	sem: *in* nm sg
fnc: **reside**	arg: **[person x]**	mdd: **reside**
mdr:	mdr: **[town y]**	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: 26	prn: 26	prn: 26

The subject/predicate relation is coded by the feature [fnc: reside] in the *[person x]* proplet and the feature [arg: [person x]] in the *reside* proplet (bidirectional pointering), and similarly for the modifier|modified relation between *reside* and *[town y].*

As a set, the proplets of a DBS content are order-free. This supports storage in, and retrieval from, a DBS database, which consists (i) horizontally of proplets with the same core value in the time-linear order of arrival, called *token lines*, and (ii) vertically in a *column of token lines* in the alphabetical order induced by the core values:

3.2 SORTING THE PROPLETS OF 3.1 INTO DBS DATABASE SCHEMA

member proplets		*now front*	*owner values*
...	sur: reinhard sanders noun: [person x] cat: snp sem: nm m sg fnc: reside mdr: nc: pc: prn: 26		[person x] = reinhold sanders
...	sur: verb: reside cat: #n′ #mdr′ decl sem: arg: [person x] mdr: [town y] nc: pc: prn: 26		reside
...	sur: schwabach noun: [town y] cat: snp sem: *in* nm sg mdd: reside mdr: nc: pc: prn: 26		[town y] = schwabach

The number of (a) member proplets in a token line and of (b) token lines is unrestricted.

4 **The On-Board Orientation System** (OBOS)

The change from a sign-based substitution-driven to an agent-based data-driven ontology leads naturally to two fundamental innovations: (i) the *on-board orientation system* (OBOS)[1] and (ii) the *now front*. Both are part of the agent's cognition, but OBOS complements the DBS database from the outside, while the now front is the processing arena within the on-board database.

One purpose of the OBOS is the interpretation of indexicals, the other the type-token distinction from philosophy. The OBOS is defined as a proplet called the STAR, for Space, Time, Agent (speaker), and Recipient (intended hearer).

4.1 EXAMPLE OF A STAR PROPLET

$$\begin{bmatrix}\text{S: yard}\\ \text{T: 2007-04-05T14:30}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 39}\end{bmatrix}$$

The S and the T values are the agent's current location and moment of time. The A and the R values refer to the speaker and the hearer. The A value is constant. The other values are provided by the continuous monitoring of the interface component.

A proposition without a STAR is the type of a content and represents its semantics:

4.2 TYPE OF THE PROPOSITION I saw you

$$\begin{bmatrix}\text{sur:}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{sem: sg}\\ \text{fnc: see}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: see}\\ \text{cat: \#n}'\text{ \#a}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: pro1 pro2}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: see}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}$$

In a type, the indexicals, here pro1 (I) and pro2 (you), are left 'dangling.'

A proposition connected to a STAR, in contrast, is a token belonging to the pragmatics:

4.3 TOKEN OF THE PROPOSITION I saw you

$$\begin{bmatrix}\text{sur:}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{sem: sg}\\ \text{fnc: see}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: see}\\ \text{cat: \#n}'\text{ \#a}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: pro1 pro2}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{sem:}\\ \text{fnc: see}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\overset{\textit{STAR-0 proplet of origin}}{\begin{bmatrix}\text{S: yard}\\ \text{T: thursday}\\ \text{A: sylvester}\\ \text{R: hector}\\ \text{3rd:}\\ \text{prn: 12}\end{bmatrix}}$$

The connection between a proposition and a STAR is a shared prn value, here 12; the indexical pro1 points at the A value sylvester, the indexical pro2 at the R value hector, and the indexical past at the T value thursday to indicate a preceding time.

5 Clearing the Now Front

A second innovation is the *now front*[2] of the DBS database schema. It serves as the arena for building elementary propositions. Before processing a next proposition, the now front must be cleared.[3] Clearance consists in moving the now front and the owners one step to the right into fresh memory space (*loom-like clearance*), leaving the concatenated proplets behind in the field of member proplets never to be changed, like *sediment*. Correction is limited to *adding* content, as in a diary entry.

5.1 SCHEMATIC EXAMPLE OF A TOKEN LINE

(i) member proplets				*(ii) now front*	*(iii) owner*
...	$\begin{bmatrix}\text{noun: square}\\ \ldots\\ \text{prn: 3}\end{bmatrix}$	$\begin{bmatrix}\text{noun: square}\\ \ldots\\ \text{prn: 6}\end{bmatrix}$	$\begin{bmatrix}\text{noun: square}\\ \ldots\\ \text{prn: 14}\end{bmatrix}$		square

For storage in the agent's on-board database, a proplet provided by automatic word form recognition is written to the now front in the token line of the owner[4] which equals its core value. For declarative retrieval (in contrast to retrieval by pointer), the first step is going to the owner corresponding to the sought proplet's core value (vertical) and the second step going along the token line to the sought proplet's prn value (horizontal).

Clearance of the now front is triggered when its proplets have ceased to be candidates for processing (e.g. cross-copying, absorption, etc.). This is basically the case when an elementary proposition is completed, formally indicated by the automatic incrementation of the prn value for the next proposition (NLC 13.5.1, IP∼START); exceptions are the extrapropositional operations of (i) coordination (NLC Chap. 11) and (ii) functor-argument (NLC Chap. 7; TExer Sects. 2.5, 2.6, 3.3–3.5). In these cases, the verb of the completed proposition must remain at the now front for cross-copying with the verb of the next proposition until the extrapropositional relation has been established.

Horizontally, the number of proplets in a token line affected by a clearance is either zero or one.[5] Vertically, over the whole column, the number of proplets at the current now front is usually no more than four or five.

The database schema, called the A-memory (formerly word bank) of DBS, is *content-addressable* for the following reasons. First, it does not use a separate index (catalog), as in a coordinate-addressable database like an RDBMS (CC Sect. 15.6; CLaTR Sect. 4.1). Second, the 'content' used for a proplet's storage in and retrieval from of the artificial agent's content-addressable memory is the letter sequence of the proplet's core value (and not a location), which enables string search (CC Sect. 12.5). Third, a core value like square is grounded in science, here geometry (FoCL 3.3.1).

[1]The OBOS may be seen as an agent-based development of Montague's (1973) sign-based index $\mathcal{M}$, i, j, g, superscripted to formulas and referring to a possible model.

[2]The now front made it possible to replace the rule packages of earlier LAG, by data-driven application: in the hear mode, a next word proplet (i) is stored in its token line at the now front prior to processing and (ii) activates all operations matching it with their second input pattern; the activated operations (iii) look for a proplet at the now front matching their first input pattern, and (iv) apply if they find one (CC Sect. 2.2).

[3]For step by step derivations of now front states see CLaTR Sect. 13.3; NLC Sects. 11.2, 11.3. For detailed derivations of the hear and the speak mode of 24 linguistically informed examples see TExer3.

[4]The terminology of member proplets and owner values is reminiscent of the member and owner records in a classic network database (Elmasri and Navathe (1989^1–2017^7), which inspired the database schema of the A-memory in DBS.

[5]Except for those rare cases in which several proplets have the same core and prn value, as in Oh Mary, Mary, Mary! or slept and slept and slept. When a now front slot is filled, a new free slot is opened. Thus, the token line of *sleep* would contain three instances of *sleep* proplets with the same prn value at the now front. They will all be left behind during the next loom-like clearance.

6 Resonating Content 1: Coactivation by Similarity

The second basic task required of an on-board database (artificial memory) is using the content being currently processed at the agent's now front to automatically *coactivate*[6] related contents stored in memory. In DBS, coactivated content is called *resonating content* and provides the basis for associations which may trigger inferences. There are three methods of coactivation: (1) similarity, (2) intersection, and (3) navigation.

Contents are computationally similar if they match the same pattern (CC Sect. 14.1). Degrees of similarity vary with the degree of pattern abstraction. Abstraction in proplets is systematically controlled by (i) the replacement of constants with variables and (ii) the restrictions on the variables (CC Sect. 14.2). Patterns of increasing abstraction degrees coactivate proplet sets of increasing size in which set n is contained in set n+1.

7 Resonating Content 2: Coactivation by Token Line Intersection

The second method of coactivating content in the on-board database is intersecting two or more token lines to find instances of the same semantic relation between similar proplets in different propositions. Derived from concept proplets at the current now front, search patterns are moved along their token lines from right to left (backwards in time). It works like a dragnet, pulled automatically (CLaTR Sect 5.1) by core values of semantically related content proplets.

Consider a search pattern derivation from two proplets in a 2nd degree intersection:

7.1 Derivation of two search patterns

$$
\begin{array}{ccc}
\textit{search patterns} & & \textit{proplets at now front} \\
\begin{bmatrix} \text{adj: big} \\ \text{mdd: square} \\ \text{prn: K} \end{bmatrix} & \Leftarrow & \begin{bmatrix} \text{sur:} \\ \text{adj: big} \\ \text{cat: adnv} \\ \text{sem: pad} \\ \text{mdd: square} \\ \text{mdr:} \\ \text{nc: blue} \\ \text{pc:} \\ \text{prn: 97} \end{bmatrix} \\
\begin{bmatrix} \text{sur:} \\ \text{noun: square} \\ \text{mdr: big} \\ \text{prn: K} \end{bmatrix} & \Leftarrow & \begin{bmatrix} \text{sur:} \\ \text{noun: square} \\ \text{cat: snp} \\ \text{sem: indef sg} \\ \text{fnc: find} \\ \text{mdr: big} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 97} \end{bmatrix}
\end{array}
$$

In this example, the search patterns express the modifier|modified relation between big and square. As they are moved in parallel along their token lines (NLC Sect. 5.1) of a DBS database, they retrieve pairs of proplets connected by (i) a shared prn value and (ii) the same semantic relation, as in the following example:

[6]So far, coactivation as a technical term seems to be confined to muscle action in neurology.

7.2 2ND DEGREE INTERSECTION COACTIVATING big square

member proplets			*now front*	*owners*
sur: adj: big cat: adn sem: pad mdd: house mdr: nc: red pc: prn: 11	sur: **adj: big** cat: adn sem: pad **mdd: square** mdr: nc: blue pc: **prn: 23**	sur: adj: big cat: adn sem: pad mdd: chair mdr: nc: pc: prn: 32	sur: adj: big cat: adn sem: pad mdd: square mdr: nc: pc: prn: 97	big
sur: **noun: square** cat: snp sem: indef sg fnc: find **mdr: big** nc: pc: **prn: 23**	sur: noun: square cat: snp sem: indef sg fnc: own mdr: green nc: pc: prn: 45	sur: noun: square cat: snp sem: indef sg fnc: buy mdr: red nc: pc: prn: 66	sur: noun: square cat: snp sem: indef sg fnc: find mdr: big nc: pc: prn: 97	square

The big|square relation retrieved has the prn value 23, in contrast to the prn value 97 of the trigger at the now front.

An intersection of two token lines is of degree 2, of three token lines of degree 3, and so on. The following example derives a search patterns for a 3rd degree intersection:

7.3 SEARCH PATTERNS FOR A 3RD DEGREE INTERSECTION

search patterns		*proplets at now front*
adj: big mdd: square prn: K	⇐	sur: adj: big cat: adnv sem: pad mdd: square mdr: nc: blue pc: prn: 97
verb: find arg: α[7] square prn: K	⇐	sur: verb: find cat: #n-s3′ #a′ decl sem: past arg: [person x] square mdr: nc: pc: prn: 97
noun: square fnc: find mdr: big prn: K	⇐	sur: noun: square cat: sn sem: sg fnc: find mdr: big nc: pc: prn: 97

For illustrating a 3rd degree search with the 7.3 patterns, the DBS database sketch 7.2 needs to be extended:

[7] For further discussion of autonomous navigation for the coactivation of content see CC Chap. 13.

7.4 RESONATING CONTENT RESULTING FROM A 3RD DEGREE SEARCH

member proplets			*now front*	*owners*
sur:	sur:	sur:	sur:	
adj: big	**adj: big**	adj: big	adj: big	big
cat: adn	cat: adn	cat: adn	cat: adn	
sem: pad	sem: pad	sem: pad	sem: pad	
mdd: house	**mdd: square**	mdd: chair	mdd: square	
mdr:	mdr:	mdr:	mdr:	
nc:	nc: blue	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 11	**prn: 23**	prn: 32	prn: 97	
sur:	sur:		sur:	
verb: find	**verb: find**		verb: find	find
cat: #n-s3′ #a′ decl	cat: #n-s3′ #a′ decl		cat: #n-s3′ #a′ decl	
sem: past	sem: pres		sem: past	
arg: [person y] triangle	arg: [person z] **square**		arg: [person x] square	
mdr:	mdr:		mdr:	
nc:	nc:		nc:	
pc:	pc:		pc:	
prn: 17	**prn: 23**		prn: 97	
sur:	sur:	sur:	sur:	
noun: square	noun: square	noun: square	noun: square	square
cat: snp	cat: snp	cat: snp	cat: snp	
sem: indef sg	sem: indef sg	sem: indef sg	sem: indef sg	
fnc: find	fnc: own	fnc: buy	fnc: sell	
mdr: big	mdr: green	mdr: red	mdr: big	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 23	prn: 45	prn: 66	prn: 97	

Compared to the number of proplets in complete token lines, the number of resonating proplets in an intersection is (i) greatly reduced and (ii) more precisely adapted to the agent's current now front content. For more precision, the number of intersections is increased. For more generality, core values provided by the now front may be replaced with more general terms in the associated semantic field hierarchies.

8 Resonating Content 3: Coactivation by Navigation

The third method of coactivating content resembles Quillian's (1968) *spreading activation*, but is more constrained in that it is restricted to navigating along existing semantic relations between proplets in the artificial agent's on-board database. Activating proplets in memory which correspond to the triggers at the current now front and following their semantic relations intra- and extrapropositional to *explore the neighborhood* has the potential of supplementing relevant information for further reasoning. Like intersection, coactivation by navigation is based technically on the database schema and the data structure of DBS:

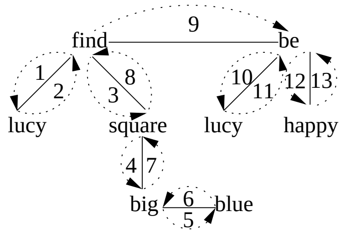
8.1 CONTENT SUPPORTING INTRAPROPOSITIONAL ACTIVATION

sur: lucy	sur:	sur:	sur:	sur:
noun: **[person x]**	verb: **find**	noun: **square**	adj: **big**	adj: **blue**
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: adnv	cat: adnv
sem: nm f	sem: pres	sem: def sg	sem: pad	sem: pad
fnc: **find**	arg: **[person x] square**	fnc: **find**	mdd: **square**	mdd:
mdr:	mdr:	mdr: **big**	mdr:	mdr:
nc:	nc: (be 24)	nc:	nc: **blue**	nc:
pc:	pc:	pc:	pc:	pc: **big**
prn: 23	prn: 23	prn: 23	prn: 23	prn: 23

The values in bold face are either core or continuation values and specify the semantic relations of structure between the proplets derived in CC 2.1.3.

Once a coactivation has resulted in the traversal of a proposition, it may continue to the next, as in Lucy found a big blue square. She was happy.:

8.2 COACTIVATION MOVING FROM ONE PROPOSITION TO THE NEXT



This content is defined as the following set of proplets connected by address:

8.3 CONTENT SUPPORTING EXTRAPROPOSITIONAL COACTIVATION

sur: lucy	sur:	sur:	sur:	sur:
noun: [person x]	verb: find	noun: square	adj: big	adj: blue
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: adnv	cat: adnv
sem: nm f	sem: pres	sem: def sg	sem: pad	sem: pad
fnc: find	arg: [person x] square	fnc: find	mdd: square	mdd:
mdr:	mdr:	mdr: big	mdr:	mdr:
nc:	nc: **(be 24)**	nc:	nc: blue	nc:
pc:	pc:	pc:	pc:	pc: big
prn: 23	prn: 23	prn: 23	prn: 23	prn: 23

sur: lucy	sur:	sur:	
noun: [person x]	verb: **be**	adj: happy	
cat: snp	cat: #n′ #be′ decl	cat: adn	
sem: nm f	sem: pres	sem: pad	
fnc: be	arg: [person x]	mdd: be	. . .
mdr:	mdr: happy	mdr:	
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 24	prn: 24	prn: 24	

The extrapropositional coordination is coded by the next conjunct feature [nc: (be 24)] of the predicate *find.*

9 Memory-Based Concatenation in Nonlanguage Recognition

The elementary features provided by the interface component of a cognitive agent may be combined in many different ways, in different modalities, and at different levels of complexity, creating a huge search space. In natural language communication this problem is solved efficiently by (i) the time-linear order of the *surface input* in the hear mode and (ii) the time-linear navigation along the semantic relations in the *content input* of the speak mode. They result in the linear degree of computational complexity in DBS (TCS), which effectively provides for real time processing.

In contrast, the cognitive processing of raw data input and output without a concrete explicit specification of the processing order, as in vision, requires alternative means of reducing the search space. In humans, they are of a cultural nature, namely the agents' knowledge of their current environment, for example, being in the kitchen, in the bath room, in a lecture hall, on the street, in a car, in a garden, in a laboratory, watching a base ball game, a Western, a sci fi movie, etc., which (i) are continously acquired by the members of a society and (ii) effectively constrain the set of associated visual concepts and their literal meaning. For example, if the agent is in the kitchen, there is normally no need to prepare for recognizing a can of motor oil.

The same restrictions are used in natural language communication, though to a much lesser extent. For example, the word match may be used in the context of lighting a fire or combining pieces of garment, bank for a financial institution, the territory alongside a river, or a place to sit. Compared to vision without domain restrictions, the number of language-dependent lexical ambiguities is tiny, but they are disambiguated in the same way, namely by the agents' awareness of the context of use in the culture.[8]

The number of distinctions within a cultural context of use depends, for example, on the agent's gender, age, origin, and education. Also, what looks the same for a layman may be obviously different, even critical, for the expert. Just as there are natural experts and natural laymen in a certain field of expertise, there may be artificial counterparts which all have the same visual equipment, but vary vastly in their cultural and scientific knowledge, and the skill of their actions.

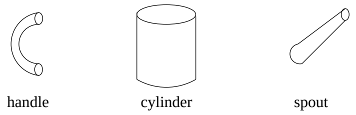
While finer and finer cultural distinctions in a domain and a semantic field are an important ingredient of better and better visual recognition, they cannot suffice alone. At the bottom level, vision must be grounded in science, just like the concepts of natural language. This raises the question of how to get from the most elementary features like the length and orientation of straight lines (Hubel and Wiesel 1962), the size and color of circles, ovals, and rectangles, etc., to cups, pails, and watering cans, for example.

In cognitive psychology, this question has been addressed by Biederman's (1987) *Recognition-by-Components* (RCB).[9] It is summarized by Kirkpatrick (2001) as follows:

> The major contribution of RBC is the proposal that the visual system extracts geons (or geometric ions) and uses them to identify objects. Geons are simple volumes such as cubes, spheres, cylinders, and wedges. RBC proposes that representations of objects are stored in the brain as structural descriptions. A structural description contains a specification of the object's geons and their interrelations (e.g., the cube is above the cylinder).
> ...
> The RBC view of object recognition is analogous to speech perception. A small set of phonemes are combined using organizational rules to produce millions of different words. In RBC, the geons serve as phonemes and the spatial interrelations serve as organizational rules. Biedermann 1987 estimated that as few as 36 geons could produce millions of unique objects.

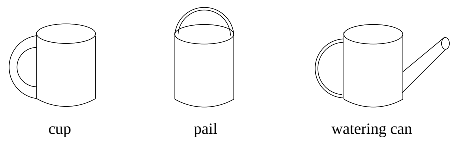
Consider the following examples of geons:

9.1 A SMALL SET OF GEONS



The geons may be assembled into concepts for complex objects such as the following:

9.2 COMBINING THE GEONS INTO MORE COMPLEX OBJECTS

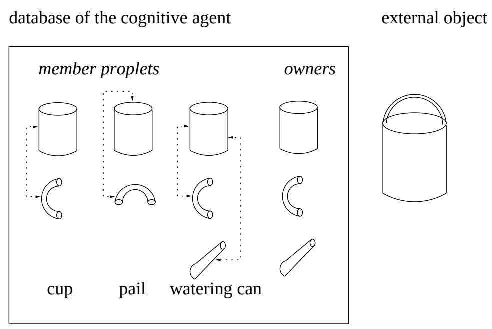


As complex objects, they raise the following question for DBS: How should the combinations of geons into concepts for complex objects be stored? For example, should the handle and the cylinder of a pail be represented adjacent, as in the graphical representation 9.2, or should we specify their connection in a more abstract manner, thus opening the way to use geons like proplets as the basic key for storage and retrieval in a DBS on-board database?

This question bears on an important task in visual recognition, namely *pattern completion* (Barsalou 1999). If visual recognition is incremental, such that we see some part first and then rapidly reconstruct the rest of the object by looking for known connections stored in memory, how can we get the database to provide relevant visual content fast and succinct to quickly narrow down the search?

It turns out that the data structure and the associated retrieval algorithm of DBS provide a highly efficient procedure of pattern completion. As an example, consider recognition of a pail based on a DBS A-memory containing the following geons:

9.3 PATTERN COMPLETION DURING RCB RECOGNITION



The isolated geons are the owners and the connected geons the members. The connections between the geons of each complex concept are indicated by dotted arrows. The complex objects specified in each column of connected geons are provided with names, i.e. cup, pail, and watering can.

Given that the external object is a pail, the agent might either first recognize the handle or the cylinder – depending on the conditions of lighting or the orientation of the object. If the cylinder is recognized first, the agent's database will indicate that cylinders are known to be connected in certain ways to handles and/or spouts. This information is used to actively analyze the cylinder's relations to the rest of the external object (pattern completion), checking for the presence or absence of the items suggested by the data base.

[8]In transgressive art, this is made aware by Meret Oppenheim's *fur cup* (1936) and in pop art by Claes Oldenburg's oversized *plantoir* (2001).

[9]Thanks to Professor Brian MacWhinney for helpful suggestions and a three month visit at the CMU Psychology Department in 1989.

Similarly, if the handle is recognized first, the agent's database indicates that handles are known to be connected in certain ways to cylinders to form cups, pails, or watering cans. If the handle-cylinder connection is recognized first, there are two possibilities: pail or watering can. In our example, the system determines that there is no spout and recognizes the object as a pail.

For specifying the connections between geons more precisely let us replace the intuitive graphical model illustrated in 9.3 with the DBS memory format based on proplets. The following example expresses the same content, but represents geons by names and codes the connections between geons by means of features (attribute value pairs).

9.4 Storing complex objects as geon proplets

members connected geons			*owners* isolated geons
geon: cylinder o: vertical a: handle back nm: cup	geon: cylinder o: vertical a: handle above nm: pail	geon: cylinder o: vertical a: handle back front spout nm: watering can	geon: cylinder orientation: attach: nm:
geon: handle o: vertical a: cylinder front nm: cup	geon: handle o: horizontal a: cylinder below nm: pail	geon: handle o: vertical a: cylinder below nm: watering can	geon: handle orientation: attach: nm:
		geon: spout o: diagonal a: cylinder behind nm: watering can	geon: spout orientation: attach: nm:

The attribute o stands for orientation, here *horizontal, vertical*, and *diagonal*, the attribute a for attachment, here *above, below, back,* and *front*, and the attribute nm for name, here *cup, pail*, and *watering can*.

In this small example with the owner geons cylinder, handle, and spout, the recognition algorithm works as follows. If the owner, for example cylinder, is matched by the raw data (data-driven activation), the algorithm checks its token line and finds three *cylinder* proplets with different nm values, namely cup, pail, and watering can, different a values, namely handle back, handle above, and handle above, spout front, and the o values vertical, horizontal, and diagonal.

The *cup* proplet has the attachment value handle back. If the interface component analysis of the raw data confirms that the handle attaches to the back of the cylinder geon and there is no spout, recognition is complete. Otherwise the attachment value handle above of the *pail* proplet is checked. If it is found in the raw data to attach at the top of the cylinder geon, recognition is complete.

The *watering can* hypothesis is the worst case, because it is alphabetically last (assuming an 'alphabetical' ordering of attachment attempts) and requires two matching checks. If none of the three hypotheses is confirmed, the recognition attemps fail and an additional complex concept may have to be constructed.

In more general applications, efficiency may be improved by using token lines with a frequency-based order in combination with domain and semantic field restrictions. Also, raw data need not always be thoroughly recognized. For example, when looking for the watering can in the context of a garden shed much of the raw data may be left unanalyzed or be analyzed crudely. By recording the raw data, however, there may be recognition from memory at a later time (as a detective's reasoning in a crime novel).

10 Conclusion

The data structure, database schema, and algorithm of a conventional database, e.g. an RDBMS, is *record-based*, while those of a DBS database serving as an artificial on-board memory are *grammar-based*. The input-output conditions of a conventional database are for storing and retrieving data, while those of a DBS database are for transfer of content from speaker to hearer in natural language communication.[10]

To show convergence[11], the paper presents two applications which go beyond the fully automatic transfer of content solely by means of raw data. One is the coactivation of resonating content in the on-board database activated by current content at the agent's now front. The other is a non-language application, namely concatenation in vision using Biederman's (1987) 'Recognition-by-Components' (RBC).

[10]In an initial response to the grammar-based approach of DBS, database experts at the University Erlangen-Nürnberg pointed out that it can be simulated by the record-based approach. As proof, the Java implementation of DBS by Arcadius Kycia (Kycia 2004), student of Erlangen Computerlinguistics, was reprogrammed by Wolfgang Fischer, student of Erlangen Informatik, as a RDBMS (Fischer 2002). To make the comparison possible, the different input-output constellations of DBS and RDBMS (Sect. 1) were left aside by using the laboratory set-up. It turned out, however, that making small changes to the RDBMS implementation, e.g. changing a category segment, had suddenly become prohibitively laborious, making the RDBMS solution impractical for upscaling.

[11]In science, the evolution of a theory shows convergence if it easily accommodates (i) established problems previously left aside and (ii) additional phenomena. Absence of convergence, in contrast, results in stagnation by producing ever new variants without any real progress, as described not so long ago by McCawley (1982) and Ross (1987).

Bibliography

Aho, A.V. and J.D. Ullman (1977) *Principles of Compiler Design*, Reading, Mass.: Addison-Wesley

Barsalou, L. (1999) “Perceptual symbol systems,” *Behavioral and Brain Sciences*, Vol. 22:577–660

Biederman, I. (1987) “Recognition-by-components: a theory of human image understanding,” *Psychological Review*, Vol. 94:115–147

CC = Hausser, R. (2019) *Computational Cognition, Integrated DBS Software Design for Data-Driven Cognitive Processing*, pp. i–xii, 1–237, `lagrammar.net`

Chisvin, L., and R.J. Duckworth (1992) “Content-addressable and associative memory,” p. 159–235, in M.C. Yovits (ed.) *Advances in Computer Science, 2nd ed.*, Academic Press

CLatr = Hausser R. (2011) *Computational Linguistics and Talking Robots; Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. at `lagrammar.net`)

Elmasri, R., and S.B. Navathe (2017) *Fundamentals of Database Systems, 7th ed.* (1st ed. 1989), Redwood City, CA: Benjamin-Cummings

Fischer, W. (2002) *Implementing Database Semantics as an RDBMS* (in German), Studienarbeit am Institut für Informatik der Universität Erlangen-Nürnberg (Prof. Meyer-Wegener), published as CLUE-Arbeitsbericht 7 (2004)

Hubel, D.H., and T.N. Wiesel (1962) “Receptive fields, binocular interaction, and functional architecture in the cat’s visual cortex,” *Journal of Physiology* 160:106–154

Kirkpatrick, K. (2001), “Object Recognition", in *Avian Visual Cognition*, cyberbook in cooperation with Comparative Cognitive Press

Kycia, A. (2004) *An Implementation of Database Semantics in Java*. M.A. thesis, CLUE

McCawley, J.D. (1982) *Thirty Million Theories of Grammar*, Chicago, Illinois: The University of Chicago Press

Montague, R. (1973) “The Proper Treatment of Quantification in Ordinary English,” in J. Hintikka, J. Moravcsik, P. Suppes (eds.) *Approaches to Natural Language*, Dordrecht, Reidel: 221–242

Peirce, C.S. (1903) ”Lectures on Pragmatism”, *Collected Papers* 5.171, C. Hartshorne and P. Weiss (eds. 1931–1935), Cambridge, MA: Harvard Univ. Press

Quillian, M. (1968) “Semantic memory,” in M. Minsky (ed.), *Semantic Information Processing*, p. 227–270, Cambridge, MA: MIT Press

Ross, J.R. (1987) *Infinite Syntax!*, Norwood: Ablex

TCS = Hausser, R. (1992) “Complexity in left-associative grammar,” *Theoretical Computer Science*, Vol. 106.2:283–308