Agent-Based Memory as an On-Board Database

Roland Hausser

Universität Erlangen-Nürnberg (em.)

Abstract

In the year 1945, John von Neumann laid down the basic structure of a cognitive agent as what is known today as a ‘von Neumann machine’ (vNm). The corresponding components in a DBS robot are (a) an interface component with sensors and actuators for automatic recognition and action (AIJ’01), (b) a content-addressable on-board database (CASM’17), and (c) a data-driven left-associative algorithm running in linear time (TCS’92).

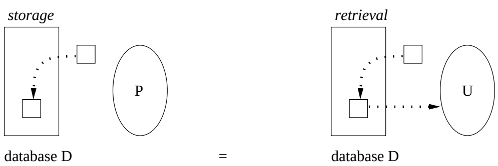
Basic tasks of a DBS robot as an artificial cognitive agent are (i) natural language communication as the transfer of content from speaker to hearer by means of raw data (1–5), (ii) association as the automatic coactivation of memory content by current processing (6–8), and (iii) efficient non-language recognition, e.g., vision (9), and action, e.g., manipulation, by an algorithm of linear-time complexity. 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 has the (i) input constellation of a programmer storing data and the (ii) output constellation of a user retrieving copies of the data:

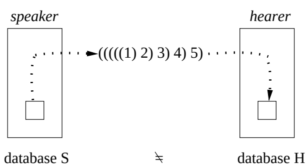
1.1 Conventional database interaction



Interaction takes place between 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. Agent P controls the storage and agent U controls the retrieval operations. Both use the same database and the same programming language, e.g., SQL (structured query language), 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 speaker and the hearer use different databases, S and 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 database schema, (ii) a data structure, and (iii) an algorithm for the storage, retrieval, and processing of content. The databases used most widely in business are the relational databases with the data structure of *records*:

2.1 RECORDS OF A RELATIONAL 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 a database. The standard operations of a record-based database,are the storage, retrieval, update, and recombining of information. For example, to retrieve the name of the representative in Schwabach, the user types the following SQL command:

2.2 DATABASE QUERY

Query:

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

Result:

```
result: A3 Sanders Reinhard
```

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

In summary, record-based databases are 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 practical advantages. The adding, finding, recombining, and correcting of 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 records, the data structure of DBS is non-recursive feature structures with ordered attributes called *proplets*, connected into content by the four semantic relations of structure, i.e., subject/predicate, object\predicate, modifier|-modified, and conjunct−conjunct, coded by address. Consider the following representation of the record A3 in 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 [fnc: reside] in the [*person x*] proplet and [arg: [person x]] in the *reside* proplet. The modifier|modified relation is coded by [mdd: reside] in the [*town y*] proplet and [mdr: [town y]] in the *reside* proplet.

The ordered attributes of the proplets are (1) the sur attribute for the optional language-dependent surface, (2) the core attribute for the obligatory values of noun, verb, or adj, (3) the cat and (4) sem attributes for syntactic-semantic information, the (5) continuation attributes fnc or arg (functor-argument), (6) the continuation attributes mdr or mdd (modifier-modified), the coordination attributes (7) nc for next conjunct and (8) pc for previous conjunct, and (9) the prn attribute for the proposition number.

As a set, the proplets of a content are order-free, which is essential for storage in and retrieval from a content-addressable database (Chisvin and Duckworth 1992). Instead of sorting data into records like name, place, customer, or employee, content-addressable DBS stores proplets based on the letter sequence of their core value, enabling computational string search. The database schema of DBS consists (i) horizontally of proplets with the same core value in the time-linear order of arrival, called token *lines*, and (ii) vertically in *columns* of token lines in the alphabetical order induced by the core values. The key is defined as the proplet address, consisting of the core and the prn value:

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] = reinhard 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 and the length of token lines are unrestricted.[1]

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

The change from a sign-based substitution-driven to an agent-based data-driven ontology lead to two fundamental innovations: (i) the *on-board orientation system*

[1]Professor Meyer-Wegener, Chair of Data Management at Erlangen Computer Science, pointed out that the DBS solution can be simulated by the record-based approach. As proof, the Java implementation of DBS by Arcadius Kycia (Kycia 2004), student at Erlangen Computerlinguistics, was reprogrammed by Wolfgang Fischer, student at Erlangen Informatik, as an RDBMS (Fischer 2002). To enable the comparison, the different input-output constellations of DBS (1.2) and RDBMS (1.1) were left aside by using the laboratory set-up (TExer 1.5). The RDBMS reconstruction succeeded, but turned out to be impractical for systematic upscaling.

and (ii) the *now front*. Both are part of the agent's cognition, but the OBOS complements the DBS database from the outside, while the now front is the processing arena inside the on-board database.

One purpose of the OBOS is the interpretation of indexicals, the other the type-token distinction from philosophy, and with it the distinction between semantics and pragmatics. The input to the OBOS are parameter values provided by the agent's interface component, the output is a continuous monitoring in form of the STAR proplet, named after the attributes Space, Time, Agent, and Recipient.

4.1 EXAMPLE OF A STAR PROPLET

S: yard
T: 2007-04-05T14:30
A: sylvester
R: speedy
3rd:
prn: 39

The S and the T values are the agent's current location and moment of time. The A, R, 3rd, and prn values refer to the speaker, hearer, 3rd person pronoun, and proposition number.[2] The A value is constant for a given agent, here sylvester, while the other values are provided by the continuous monitoring of the OBOS.

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

4.2 TYPE OF THE NONLANGUAGE CONTENT I saw you

sur:	sur:	sur:
noun: pro1	verb: see	noun: pro2
cat: s1	cat: #n′ #a′ decl	cat: sp2
sem: sg	sem: past ind	sem:
fnc: see	arg: pro1 pro2	fnc: see
...	...	...
prn: K	prn: K	prn: K

In a type, the prn values are substitution variables, here K, and there is no STAR, leaving the indexicals 'dangling,' here pro1 (I) and pro2 (you).

A proposition connected to a STAR, in contrast, is a content token and belongs to the pragmatics:

4.3 TOKEN OF THE NONLANGUAGE CONTENT I saw you

			STAR-0 proplet of origin
sur:	sur:	sur:	S: yard
noun: pro1	verb: see	noun: pro2	T: thursday
cat: s1	cat: #n′ #a′ decl	cat: sp2	A: sylvester
sem: sg	sem: past ind	sem:	R: hector
fnc: see	arg: pro1 pro2	fnc: see	3rd:
...	...	...	prn: 12
prn: 12	prn: 12	prn: 12	

The connection between a proposition and a STAR is a shared prn value, here 12;

[2]For easy pronunciation, the attributes 3rd and prn are omitted in the STAR acronym.

the indexical pro1 points at the A value sylvester, the indexical pro2 at the R value hector, and the indexical past at the present time value T, here thursday.

5 Loom-like Clearance of the Now Front

The second innovation of the DBS database schema is the *now front*[3]. Before processing a next proposition, the now front is cleared:[4]

5.1 TOKEN LINE WITH CLEAR NOW FRONT

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

Clearance consists in moving the now front with the owners into fresh memory space (*loom-like clearance*), leaving the concatenated proplets behind in the field of member proplets, never to be changed, like *sediment*. Correcting content is limited to *adding* content (change in hindsight, as in a diary entry), using reference by address.

For storage, a proplet provided by automatic word form recognition is written to the now front in the token line of the owner[5] 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 is going along the token line to the sought proplet's prn value (horizontal).

The now front is cleared when its proplets have ceased to be candidates for processing (e.g., cross-copying, absorption, activation). 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); partial exceptions are the extrapropositional operations of (i) coordination (NLC 11) and (ii) functor-argument (NLC 7; TExer 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, or navigation to, the verb of the next proposition until the extrapropositional relation has been utilized or established.

Because the proplets at the current now front are limited to an elementary proposition, their number, vertically over the whole column, is usually no more than four or five. Horizontally, the number of proplets in a token line affected by a clearance is either zero or one.[6]

[3]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 2.2).

[4]For step by step derivations of now front states see CLaTR 13.3; NLC 11.2, 11.3. For explicit declarative derivations in the hear and speak modes of 24 linguistically informed examples see TExer.

[5]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] 2017), which inspired the database schema of the A-memory in DBS.

6 Resonating Content 1: Coactivation by Similarity

An important property of natural cognition is *association* (psychology), i.e. the automatic activation of content in memory which is related to current content. The associated content enriches the literal content with individual reminiscences as well as general knowledge. In DBS, this process is modeled as the automatic *coactivation* (CC 13) of related contents stored in memory, called *resonating content*. The three method of coactivation are (1) similarity, (2) intersection, and (3) continuation.

Contents are computationally similar if they match the same pattern (CC 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) restrictions on the variables (CC 14.2). Patterns of increasing abstraction degrees coactivate 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 is intersecting two or more token lines to find instances of the same semantic relation between similar proplets in different propositions. 2nd degree search patterns derive from intersecting two proplets:

7.1 DERIVATION OF TWO SEARCH PATTERNS

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

Automatically derived from concept proplets at the current now front, search pat-

[6] 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.

terns are moved along their token lines from right to left (backwards in time).

For example, the search patterns in 7.1 express the modifier|modified relation between big and square. As they are moved in parallel along their token lines (NLC 5.1) in 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:

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

$$
\begin{bmatrix}\text{verb: find}\\ \text{arg: } \alpha^{7} \text{ square}\\ \text{prn: K}\end{bmatrix}
\Leftarrow
\begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: past}\\ \text{arg: [person x] square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}
$$

$$
\begin{bmatrix}\text{noun: square}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{prn: K}\end{bmatrix}
\Leftarrow
\begin{bmatrix}\text{sur:}\\ \text{noun: square}\\ \text{cat: sn}\\ \text{sem: sg}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}
$$

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

7.4 RESONATING CONTENT RESULTING FROM A 3RD DEGREE SEARCH

member proplets			*now front*	*owners*
sur: adj: big cat: adn sem: pad mdd: house mdr: nc: 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: verb: find cat: #n-s3′ #a′ decl sem: past arg: [person y] triangle mdr: nc: pc: prn: 17	sur: **verb: find** cat: #n-s3′ #a′ decl sem: pres arg: [person z] **square** mdr: nc: pc: **prn: 23**		sur: verb: find cat: #n-s3′ #a′ decl sem: past arg: [person x] square mdr: nc: pc: prn: 97	find
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: sell mdr: big nc: pc: prn: 97	square

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 triggers at the agent's current now front content. As the computational counterpart to associating freely in natural cognition, artificial coactivation is an important part of automated reasoning. For more precision, the number of intersections is increased. For more generality, core values provided by the now front are replaced with more general terms in the associated semantic field hierarchies.

8 Resonating Content 3: Coactivation by Continuation

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 all proplets in memory which correspond to a single trigger at the current now front and following their semantic relations intra- and extrapropositionally to 'explore the neighborhood' has the potential of providing cognition with relevant information for further reasoning. Like intersection, coactivation by continuation is based technically on the database schema and the data structure of DBS:

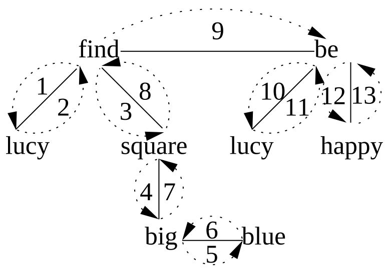
8.1 Content supporting intrapropositional activation

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* to the hear mode and (ii) the time-linear navigation along the semantic relations in the *content input* to the speak mode. They result in the linear degree of computational complexity in natural language communication, which effectively provides for real time processing in LAG/DBS (TCS'92).

In contrast, the cognitive processing of raw data input and output without an explicit specification of the processing order, for example 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 interconnections. For example, if the agent is in the well-kept garage of an electric vehicle,, 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 in the skill of their actions.

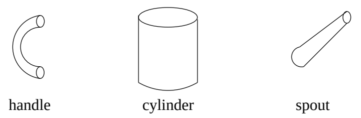
While finer and finer cultural distinctions in a domain and a semantic field are an important ingredient of better and better visual recognition, they cannot suffice alone. At the bottom level, vision must be grounded in science, for language and nonlanguage concepts alike. This raises the question of 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,[9] this question has been addressed by Biederman's (1987) *Recognition-by-Components* (RCB). 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. Biederman 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



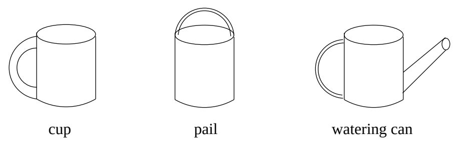
The geons may be assembled into concepts for complex objects:

---

[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.

9.2 COMBINING GEONS INTO MORE COMPLEX OBJECTS

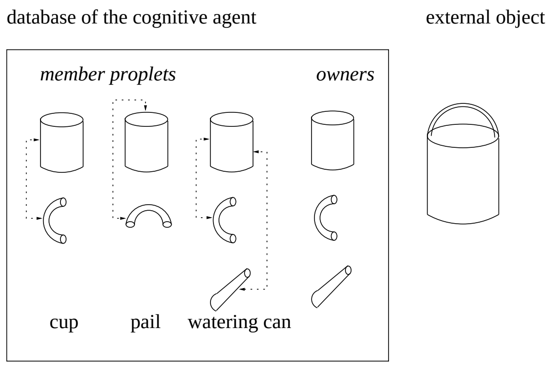


As complex objects, they raise the following question for DBS: How should the combinations of geons into concepts for complex objects be stored? For example, should the handle and the cylinder of a pail be represented adjacent, as in the graphical representation 9.2, or should we specify their connection 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, here 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.

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 a variant of 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 different o values, namely 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 rather than frequency ordering of attachment attempts) and requires two matching checks. If none of the three hypotheses is confirmed, the recognition attempts fail and an additional complex concept is 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 cluttered garden shed much of the raw data may be left unanalyzed or be analyzed crudely. By recording the raw data without analysis, however, there may be recognition from memory at a later time (such as a detective's reasoning on her sofa in front of a blazing fire with a bottle of single malt).

10 Conclusion

A relational database, e.g., an RDBMS, is *record-based* and used for storage in, and retrieval from, large collections of systematic information, such as banking, bibliographical, insurance, medical, tax, and trade data. The input-output conditions of a conventional database are for storing and retrieving data.

A DBS database, in contrast, is *content-based* and serves as the on-board memory of an artificial agent. The input-output conditions of a DBS database are for (i) automatic nonlanguage recognition and action, (ii) the transfer of content from speaker to hearer in natural language communication, and (iii) reasoning.

The psychological phenomenon of association is reconstructed computationally as the coactivation of resonating content in the on-board database, activated by content at the agent's current now front. For the concatenation of concepts without the benefit of an externally-given processing order in non-language visual perception, the memory-based retrieval mechanism of DBS is applied to Biederman's (1987) 'Recognition-by-Components'.

Bibliography

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

AIJ'01 = Hausser, R. (2001) "Database Semantics for Natural Language," *Artificial Intelligence*, 130.1:27–74, Elsevier

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

CASM'17 = Hausser, R. (2017) "A computationalreatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26, Springer

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

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

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

Neumann, J.v. (1945) *First Draft of a Report on the EDVAC*, in IEEE Annals of the History of Computing. Vol. 15, Issue 4, 1993, doi:10.1109/85.238389, pp. 27–75

NLC = Hausser, R. (2006) *A Computational Model of Natural Language Communication – Interpretation, Inferencing, and Production in Database Semantics*, pp. 360, Springer

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

TCS'92 = Hausser, R. (1992) "Complexity in left-associative grammar," *Theoretical Computer Science*, Vol. 106.2:283–308, Elsevier