Applying Database Semantics to the WWW

Roland Hausser

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

Abstract Today's search engines build their indices on the basis of document mark-up in XML and significant letter sequences (words) occurring in the document texts. There are some drawbacks, however: the XML mark-up requires skill as well as tedious work from the user posting the document, and the indexing based on significant word distributions, though automatic and highly effective, is not as precise as required by many applications.
As a complement to current methods, this paper presents an automatic content analysis of texts which is based on traditional linguistic methods in conjunction with a comparatively new data structure (AIJ'01) and algorithm (TCS'92). Having already presented the formal definitions elsewhere, we aim here at illustrating the system in action, based on an ongoing implementation in JAVA.

Database Semantics is the name of a computational system modeling natural language communication. DBS is designed for the construction of talking robots. Because of this background, applications of DBS to the internet will be presented by going from the agent's *hearer-mode* (interpretation of language) to the agent's *conceptualization* (choosing what to say and how to say it) to the agent's *speaker-mode* (production of language). Once the basic functioning of DBS has been explained, we will show how to incorporate its components into the internet and which www applications it improves.

1 Word Form Recognition in the Hearer-Mode

For the computer, the word forms in a text are merely sequences of letters. The first step towards computers *understanding* natural language is automatic word form recognition. This software component provides each word form in a sentence or text automatically with a lexical analysis.

Consider the following example of automatically recognizing the word form girl:

1.1 Example of automatic word form recognition in DBS

```
dbs2.DBS2-HEAR> -lex girl

  [sur: girl ]
  [noun: girl]
  [cat: sn   ]
  [sem: sg f ]
  [mdr:      ]
  [fnc:      ]
  [idy: +1   ]
  [prn:      ]
```

This lexical analysis has the form a feature structure, called *proplet*. Feature structures consist of a set of attribute-value pairs. Proplets are restricted, however, in that their attributes may not take feature structures as values, thus preventing a recursive embedding of feature structures in proplets.[1]

For better readability, the attributes of a proplet are displayed in a predefined standard order. The attribute sur contains the surface of the word as value. The attribute noun specifies the part of speech; its value is a concept which characterizes the core of the word meaning. The attributes cat (for category) and sem (for semantics) specify the morphosyntactic properties of the word.[2] The remaining attributes mdr (modifier), fnc (functor), idy (identity), and prn (proposition number) receive their values during syntactic-semantic parsing (see Section 2 below).

A system of automatic word form recognition provides much more detailed and reliable grammatical information than statistical tagging. Based on a finite lexicon and rules for inflection, derivation, and composition, the system recognizes a potentially infinite number of words. Furthermore, if a traditional lexicon is available on-line, a suitable system of automatic word form recognition can be built in a matter of weeks for any new language. Unlike statistical tagging, mistaken analyses can be precisely located and permanently corrected.[3] With the support of a reference corpus and a continuous sequence of monitor corpora, a system of automatic word form recognition can be maintained to provide near-perfect coverage, serving a wide range of applications.

2 Syntactic-Semantic Parsing in the Hearer-Mode

The next step in modeling the hearer-mode is syntactic-semantic parsing. It has the task of establishing the grammatical relations between the lexically analyzed word forms. In Database Semantics, grammatical relations are coded by *copying values* between proplets. Consider the following example of parsing the sentence the girl sleeps:

2.1 BUILDING THE FUNCTOR-ARGUMENT STRUCTURE

result of word form recognition ⇒ *result of syntactic-semantic parsing*[4]

```
[sur: the   ] [sur: girl ] [sur: sleeps]       [sur:         ] [sur:        ]
[noun: _n1  ] [noun: girl] [verb: sleep]       [noun: girl   ] [verb: sleep]
[cat: nn' np] [cat: sn   ] [cat: ns3' v]       [cat: snp     ] [cat: v      ]
[sem: def   ] [sem: sg f ] [sem: pres  ]       [sem: def sg f] [sem: pres   ]
[mdr:       ] [mdr:      ] [mdr:       ]       [mdr:         ] [mdr:        ]
[fnc:       ] [fnc:      ] [arg:       ]       [fnc: sleep   ] [arg: girl   ]
[idy: +1    ] [idy: +1   ] [ctn:       ]       [idy: 1       ] [ctn:        ]
[prn:       ] [prn:      ] [ctp:       ]       [prn: 1       ] [ctp:        ]
                           [prn:       ]       [wrdn: 1      ] [prn: 1      ]
                                                                [wrdn: 3     ]
```

[1] The attributes of proplets may take more than one value, however, as in a list.

[2] More specifically, the values sn stands for singular noun, sg for singular, and f for feminine.

[3] For a comparison of different approaches to automatic word form recognition see Hausser 1996 and FoCL'99, Chapters 13–15.

[4] Once the values of the sur attributes have been used for automatic word form recognition, they are discarded during syntactic-semantic parsing, resulting in a more language-independent semantic representation.

The three proplets on the left result from automatic word form recognition and serve as input to the parser. The two proplets on the right are the output of the parser.

The parser reconstructs the functor-argument structure of the sentence by copying the noun value girl into the arg attribute of the verb, and the verb value sleep into the fnc attribute of the noun (bidirectional pointering).[5] Note that the input proplets *the* and *girl* are fused into a single proplet in the output (function word absorption), whereby the contribution of *the* is reflected by the value def in the sem slot of the resulting noun proplet. Furthermore, the valency position ns3' (for nominative singular third person) in the cat slot of the verb proplet in the input is being canceled in the output.

In addition to the intrapropositional functor-argument structure of isolated sentences, Database Semantics characterizes the concatenation of sentences in a text. There are two kinds of extrapropositional relations, (i) the conjunction between verbs and (ii) the identity between nouns. The conjunction between verbs is shown by the following example, which shows of the syntactic-semantic analysis of

> The big girl has been sleeping. The young girl ate a big warm meal. The old girl is eating a big apple.

2.2 RESULT OF PARSING A SHORT TEXT

```
[sur:          ] [sur:       ] [sur:                   ] [sur:          ] [sur:       ] [sur:           ]
[noun: girl    ] [adj: big   ] [verb: sleep            ] [noun: girl    ] [adj: young] [verb: eat      ]
[cat: SNP      ] [cat: ADN   ] [cat: DECL              ] [cat: SNP      ] [cat: ADN   ] [cat: DECL      ]
[sem: def sg F] [sem: stand] [sem: hv-pres perf prog] [sem: def sg F] [sem: stand] [sem: past      ]
[mdr: big      ] [mdd: girl  ] [mdr:                   ] [mdr: young    ] [mdd: girl  ] [mdr:           ]
[fnc: sleep    ] [idy: 2     ] [arg: girl              ] [fnc: eat      ] [idy: 3     ] [arg: girl meal]
[idy: 2        ] [prn: 2     ] [ctn: 3 eat             ] [idy: 3        ] [prn: 3     ] [ctn: 4 eat     ]
[prn: 2        ] [wrdn: 2    ] [ctp:                   ] [prn: 3        ] [wrdn: 2    ] [ctp: 2 sleep   ]
[wrdn: 1       ]               [prn: 2                 ] [wrdn: 1       ]               [prn: 3         ]
                                [wrdn: 4                ]                                [wrdn: 4        ]

[sur:          ] [sur:       ] [sur:       ] [sur:          ] [sur:       ] [sur:              ]
[noun: meal    ] [adj: big   ] [adj: warm  ] [noun: girl    ] [adj: old   ] [verb: eat         ]
[cat: SNP      ] [cat: ADN   ] [cat: ADN   ] [cat: SNP      ] [cat: ADN   ] [cat: DECL         ]
[sem: indef sg] [sem: stand] [sem: stand] [sem: def sg F] [sem: stand] [sem: be-pres prog]
[mdr: big warm] [mdd: meal  ] [mdd: meal  ] [mdr: old      ] [mdd: girl  ] [mdr:              ]
[fnc: eat      ] [idy: 4     ] [idy: 4     ] [fnc: eat      ] [idy: 5     ] [arg: girl apple  ]
[idy: 4        ] [prn: 3     ] [prn: 3     ] [idy: 5        ] [prn: 4     ] [ctn:              ]
[prn: 3        ] [wrdn: 6    ] [wrdn: 7    ] [prn: 4        ] [wrdn: 2    ] [ctp: 3 eat        ]
[wrdn: 5       ]                             [wrdn: 1       ]               [prn: 4            ]
                                                                             [wrdn: 4           ]

[sur:          ] [sur:       ]
[noun: apple   ] [adj: big   ]
[cat: SNP      ] [cat: ADN   ]
[sem: indef sg] [sem: stand]
[mdr: big      ] [mdd: apple]
[fnc: eat      ] [idy: 6     ]
[idy: 6        ] [prn: 4     ]
[prn: 4        ] [wrdn: 7    ]
[wrdn: 6       ]
```

The concatenation of the three sentences in the sample text is coded by the values of the ctn (connection to next) and ctp (connection to previous) attributes of the verbs. More specifically, the ctp attribute of the verb proplet *sleep* has no value because there is no previous sentence, while the ctn attribute has the values 3 eat, representing the prn value and the verbal concept of the next proposition. Accordingly, the ctp attribute of the verb proplet *eat* has the values 2 sleep, while the ctn attribute has the values 4 drive, etc.

---

[5] Other values are provided by the control structure of the parser: the prn attributes receive a common number, the idy attribute receives a number, and the new attribute wrdn (for word number) is being added and supplied with suitable values.

Example 2.2 also shows integration of the modifiers (adjectives) big, young, and old into the functor-argument structure as attribute values. It shows the handling of transitive sentences with the objects meal and apple. And it shows the complex verb constructions has been sleeping and is eating, which are each represented by a single proplet (another instance of function word absorption, compare 2.1).

The syntactic-semantic parser producing the set of proplets in 2.2 is based on the time-linear algorithm of LA-grammar (TCS'92, FoCL'99, AIJ'01, Hausser 2005).[6] Extending the coverage of this parser is considerably more demanding than providing the lexical entries and rules for the automatic word form recognition of a natural language. This is because automatic word form recognition can be built on centuries of lexicographic work analyzing word forms grammatically, while no suitable[7] analyses are available for the grammatical relations between word forms in sentences and texts.

Nevertheless, near complete syntactic-semantic coverage of a natural language in the format shown in 2.2 can be achieved in a few years. Our strategy is as follows: Using automatic word form recognition in combination with a chunk parser (Abney 1991, Vergne 2002), the sequences of word forms in a corpus are boiled down to sequences of categories in phrases. After ordering these category sequences according to frequency and analyzing the most frequent ones first, the time-linear syntactic-semantic coverage of free text can be upscaled very efficiently.

3 Storage and Retrieval in a Word Bank

By coding the intra- and extrapropositional relations between the words in a text as attribute values, proplets are autonomous items which can be stored independently of any restrictions imposed by graphical representations such as trees. In other words, the storage of proplets is completely free and the principle for their storage can be chosen according to the needs of one's database.

In DBS, proplets are stored in the format of alphabetically ordered *token lines*. The first item of a token line is a concept; it is followed by all proplets containing this concept as the value of their second attribute. This data structure resembles a classic network database with its owner and member records, and is called a *word bank*. As an example, consider the following word bank storing the proplets derived in 2.1 and 2.2:

3.1 Illustrating the Data Structure of a Word Bank

```
Owner Proplets |-------- Token Line ---------->
---------------------------------------------------------------------------
[noun: apple] [sur:         ]
              [noun: apple  ]
              [cat: SNP     ]
              [sem: indef sg]
              [mdr: big     ]
              [fnc: eat     ]
              [idy: 6       ]
              [prn: 4       ]
              [wrdn: 6      ]
```

[6] LA-grammar is named after its **L**eft-**A**ssociative derivation order. LAG computes *possible continuations*, always combining a sentence start and a next word into a new sentence start.

[7] Most efforts in modern linguistics are not time-linear and therefore not input-output equivalent with the speaker-hearer. For further discussion see FoC'99L, Chapters 8 and 9.

```
[adj: big] [sur:      ] [sur:      ] [sur:      ]
           [adj: big  ] [adj: big  ] [adj: big  ]
           [cat: ADN  ] [cat: ADN  ] [cat: ADN  ]
           [sem: stand] [sem: stand] [sem: stand]
           [mdd: girl ] [mdd: meal ] [mdd: apple]
           [idy: 2    ] [idy: 4    ] [idy: 6    ]
           [prn: 2    ] [prn: 3    ] [prn: 4    ]
           [wrdn: 2   ] [wrdn: 6   ] [wrdn: 7   ]

[verb: eat] [sur:           ] [sur:              ]
            [verb: eat      ] [verb: eat         ]
            [cat: DECL      ] [cat: DECL         ]
            [sem: past      ] [sem: be-pres prog]
            [mdr:           ] [mdr:              ]
            [arg: girl meal] [arg: girl apple   ]
            [ctn: 4 eat     ] [ctn:              ]
            [ctp: 2 sleep   ] [ctp: 3 eat        ]
            [prn: 3         ] [prn: 4            ]
            [wrdn: 4        ] [wrdn: 4           ]

[noun: girl] [sur:          ] [sur:          ] [sur:          ] [sur:          ]
             [noun: girl    ] [noun: girl    ] [noun: girl    ] [noun: girl    ]
             [cat: SNP      ] [cat: SNP      ] [cat: SNP      ] [cat: SNP      ]
             [sem: def sg F] [sem: def sg F] [sem: def sg F] [sem: def sg F]
             [mdr:          ] [mdr: big      ] [mdr: young    ] [mdr: old      ]
             [fnc: sleep    ] [fnc: sleep    ] [fnc: eat      ] [fnc: eat      ]
             [idy: 1        ] [idy: 2        ] [idy: 3        ] [idy: 5        ]
             [prn: 1        ] [prn: 2        ] [prn: 3        ] [prn: 4        ]
             [wrdn: 1       ] [wrdn: 1       ] [wrdn: 1       ] [wrdn: 1       ]

[noun: meal] [sur:          ]
             [noun: meal    ]
             [cat: SNP      ]
             [sem: indef sg]
             [mdr: big warm]
             [fnc: eat      ]
             [idy: 4        ]
             [prn: 3        ]
             [wrdn: 5       ]

[adj: old] [sur:      ]
           [adj: old  ]
           [cat: ADN  ]
           [sem: stand]
           [mdd: girl ]
           [idy: 5    ]
           [prn: 4    ]
           [wrdn: 2   ]

[verb: sleep] [sur:        ] [sur:                    ]
              [verb: sleep] [verb: sleep             ]
              [cat: DECL  ] [cat: DECL               ]
              [sem: pres  ] [sem: hv-pres perf prog]
              [mdr:        ] [mdr:                    ]
              [arg: girl  ] [arg: girl               ]
              [ctn:        ] [ctn: 3 eat             ]
              [ctp:        ] [ctp:                    ]
              [prn: 1      ] [prn: 2                 ]
              [wrdn: 3     ] [wrdn: 4                ]

[adj: warm] [sur:      ]
            [adj: warm ]
            [cat: ADN  ]
            [sem: stand]
            [mdd: meal ]
            [idy: 4    ]
            [prn: 3    ]
            [wrdn: 7   ]

[adj: young] [sur:      ]
             [adj: young]
             [cat: ADN  ]
             [sem: stand]
             [mdd: girl ]
             [idy: 3    ]
             [prn: 3    ]
             [wrdn: 2   ]
```

The reordering of a proplet sequence produced by the parser, e.g. 2.2, into the word bank format, e.g. 3.1, is automatic.

The word bank format has the advantage of easy storage and retrieval. The storage of a new sentence consists in adding its proplets at the end of their token lines. For example, a new proplet *girl* would be added at the end of the girl token line. The proplets in a token line reflect the temporal order of their arrival, by position and their prn number.

The retrieval of proplets is based on their concept value, their proposition number (prn), and their word number (wrdn), which jointly serve as the unique primary key. Proplet retrieval is needed by the following operations:

3.2 OPERATIONS USING PROPLET RETRIEVAL

1. *Internal navigation*
   For any proplet, a successor proplet may be retrieved. Repetition of this operation results in a time-linear navigation through the content of the word bank.
2. *External activation*
   For any concept provided by recognition, all proplets corresponding to this concept are retrieved by activating the concept's token line.

Both kinds of operations result in a selective activation of the content in a word bank.

The first type of operation is used to activate a sequence of propositions. For example, if we pick the second *eat* proplet in 3.1, the first arg value and the prn value tell the system to retrieve the *girl* proplet with the prn value 4. After returning to the verb, the second arg value is used to navigate to *apple*. After returning to the verb, the ctp value is used to navigate to the verb of the previous proposition, and so on.

The second type of operation is used to answer questions. For example, to answer the question *Which girl ate the meal?* based on the word bank 3.1, the system searches the token line of *eat* from right to left (i.e. going back in time), looking for the arg values *girl* and *meal*. The matching *eat* proplet has the prn value 3. Next, the system retrieves the *girl* proplet with the prn value 3 (first kind of 3.2 operation). Based on its mdr and prn values, the system determines that it was the *young girl* who ate the meal.[8]

This kind of retrieval is much more precise than a comparable full text search. For example, a Google search with the words *girl eat meal* currently results in 433 000 sites like the following:

> ... In fact, I had way too many for the whole of Mexico to eat. ... They make good apple sauce and they don't cost a dime. ... 2000-2003 You Grow Girl & Fluffco. ..

4 Semantic-Syntactic Parsing in the Speaker-mode

In DBS, the production of language is based on an autonomous navigation through the content of the word bank. The navigation uses the relations between the proplets as a railroad system and an LA-grammar, called **LA-think**, as the motor-algorithm for moving the word bank's unique focus point along the rails.

Given that proplets usually provide more than one possible successor proplet, the system must make choices. The most basic solution are random choices. For rational behavior, however, the **LA-think** grammar must be refined into a control structure which chooses between continuation alternatives based on the evaluation of external and internal stimuli, the frequency of previous traversals, learned procedures, theme/rheme structure, etc.

[8] For a more formal description see AIJ'01.

Alternative navigations through the same propositional content are illustrated by the following set of proplets. It was generated automatically by **LA-hear** interpreting the sentence sequence The girl left the house. Then the girl crossed the street.

4.1 PROPLETS OF *girl leave house. girl cross street*

```
[sur:         ] [sur:            ] [sur:         ]
[noun: girl   ] [verb: leave     ] [noun: house  ]
[cat: SNP     ] [cat: DECL       ] [cat: SNP     ]
[sem: def sg F] [sem: past       ] [sem: def sg  ]
[mdr:         ] [mdr:            ] [mdr:         ]
[fnc: leave   ] [arg: girl house ] [fnc: leave   ]
[idy: 1       ] [ctn:            ] [idy: 2       ]
[prn: 1       ] [ctp: 1 leave    ] [prn: 1       ]
[wrdn: 1      ] [prn: 1          ] [wrdn: 4      ]
                [wrdn: 3         ]

[sur:         ] [sur:            ] [sur:         ]
[noun: girl   ] [verb: cross     ] [noun: street]
[cat: SNP     ] [cat: DECL       ] [cat: SNP     ]
[sem: def sg F] [sem: past       ] [sem: def sg  ]
[mdr:         ] [mdr:            ] [mdr:         ]
[fnc: cross   ] [arg: girl street] [fnc: cross   ]
[idy: 1       ] [ctn:            ] [idy: 3       ]
[prn: 2       ] [ctp: 1 leave    ] [prn: 2       ]
[wrdn: 1      ] [prn: 2          ] [wrdn: 4      ]
                [wrdn: 3         ]
```

This data structure may be traversed forward, as in
girl leave house | *girl cross street*
which is reflected by the English surface
The girl left the house. Then she crossed the street.
It may also be traversed backwards, as in
girl cross street | *girl leave house*
which is reflected by the English surface
The girl crossed the street. Before that she left the house.
Furthermore, using the identity between the two *girl* proplets,[9] the navigation may enter the second proposition before finishing the first, as in
girl | *girl leave house* | *cross street*,
reflected in English by the relative clause construction
The girl who left the house crossed the street.
Using the conjunction between the verb proplets, the navigation may also produce adverbial subclauses such as
After the girl left the house, she crossed the street. (forward)
Before the girl crossed the street, she left the house. (backward)
An intrapropositional kind of backward navigation is passive, as in
The street was crossed by the girl.
Each of these traversals is based on a particular rule sequence of **LA-think**.[10]

The choice between these different traversals is motivated by how the agent views the content, i.e. from where the navigation enters the content (forward or backward), which

[9] Because identity-inference has not yet been implemented, the idy values of the two *girl* proplets were set to equal by hand.
[10] For explicit definitions of these rules see FoCL'99, AIJ'01, Hausser 2005.

part is evaluated as foreground and which as background, etc. In addition to navigation merely activating the content contained in the word bank, there is also navigation which produces new content, called inference navigation. Like all kinds of navigation, inference navigation is controlled by the rules of a suitable **LA-think** grammar.

5 Word Form Production in the Speaker-Mode

A computational model of the speaker-mode raises the question of where the content to be uttered should come from. Intuitively, the answer is obvious: it should come from thought. But how should thought be modeled?

In DBS, thought is modeled as the time-linear navigation through the wordbank, controlled by LA-grammars for activating content, evaluating recognition, initiating action, planning, drawing conclusions, etc. In principle, any such navigation through the word bank is independent of language. However, in cognitive agents with language, the navigation serves as the speaker's *conceptualization*, i.e., as the speaker's choice of *what to say* and *how to say it.*

A conceptualization defined as a time-linear navigation through content makes language production relatively straightforward: If the speaker decides to communicate a navigation to the hearer, the concept names (i.e., values of the second attributes) of the proplets traversed by the navigation are translated into their language-dependent counterparts and realized as external signs. For example, traversing the proplets

eat girl young meal big warm

would result in the following surface sequences, depending on the language:

English: eat girl young meal big warm
French: mange fille jeune repas copieux chaud
German: essen Mädchen jung Mahlzeit groß warm
Korean: mek.ta so.nye e.ri.ta um.sik manh.ta tta.ttus.ha.ta

In addition to this language-dependent lexicalization of the universal navigation, the system must provide

1. language-dependent word order
2. function word precipitation
3. word form selection for proper agreement

For example, each of the above base form sequences must be mapped into

English: the young girl ate a big warm meal
French: la jeune fille mangait un copieux repas chaud
German: das junge Mädchen aß eine große warme Mahlzeit
Korean: e.ri-n so.nye-ka tta.ttus.ha-n um.sik-ul manh-i mek-nun-ta

This process is handled by language-dependent LA-grammars, called **LA-speak**, in combination with language-dependent word form production systems. For example, the English word form ate is produced from an *eat* proplet the sem attribute of which contains the value past.

Given the time-linear derivation order common to **LA-hear, LA-think,** and **LA-speak**, one would hope that semantic-syntactic parsing and automatic word form production (speaker-mode) would reuse much of what has been built for automatic word form recognition and syntactic-semantic parsing (hearer-mode). The tasks of the two modes are quite different, however.

For example, automatic word form recognition disassembles a given letter sequence into a sequence of lexically analyzed allomorphs which are reassembled into a proplet, while automatic word form production selectively matches the values of a given proplet or set of proplets with lexically analyzed allomorphs which are assembled into a suitable surface. Nevertheless, the goal for designing the interpretation and the production system for a language must be to utilize the same data as much as possible.

6 Applying Database Semantics to the World Wide Web

The system of Database Semantics described above is implemented in Java as a prototype handling fragments of English, German, and Korean in the speaker- and the hearer-mode. Because there are presently no suitable robots available, DBS runs on standard computers – which limits recognition and action to the language level.

Without the technology of robots recognizing their environment and acting in it autonomously, concepts cannot be defined in terms of recognition and action procedures, and thus are not available to be reused as the core of lexical meanings (word semantics). This happens to be a serious, though hopefully temporary, deficit for the software design of autonomous cognitive agents communicating in natural language. For the transfer of Database Semantics to the internet, however, it presents no obstacle. Consider the following www applications:

LA-hear grammars improve precision by parsing www documents, representing their content in the uniform format of a word bank (cf. 3.1). All that is required to utilize this giant index structure for document retrieval is one additional proplet attribute, called URL, for specifying the location of the proplet's document.

In this word bank, noun proplets specify their functors, verb proplet specify their arguments, modifier proplets specify their modified and vice versa, nouns specify identity with other nouns, and propositions specify their predecessor and and successor. For example, if the user is looking for documents on *professors driving BMWs*, the system would search through the token line of *drive*, collect all *drive* proplets with the arguments *professor* and *BMW*, and retrieve the associated documents by using the value of the proplets' URL attribute. In comparison, a free text search in Google using the words *professor drive BMW* currently returns 19 000 sites such as:

> ... Developing a hybrid vehicle that drives for long ... a realistic development proposition," emphasizes Professor Göschel. BMW has already played through an extreme ...

LA-think grammars improve recall by inferencing. For example, a search for *professors driving German cars* would initially overlook *professors driving BMWs*. However, based on the absolute propositions *BMWs are German cars, Mercedes are German cars, Porsches are German cars*, etc., the system infers that documents on professors driving BMWs are relevant for a query on professors driving German cars.

In addition, **LA-think** grammars can be developed into what Berners-Lee et al. 2001 envisage as agents roaming the net to perform jobs for individual users. However, instead of being based on the railroad system of RDF, hand-coded in XML for each document, web agents based on **LA-think** use the relations between proplets, established automatically by **LA-hear**. This kind of web application may utilize **LA-think** inferences developed independently for artificial cognitive agents as well as foster its own.

LA-speak grammars, finally, map content resulting from inferencing and represented as a sequence of proplets into natural language. Thus, in a web extended to DBS, the response to a query would not only be the retrieval of thousands of documents, but a selectively derived to-the-point answer using various resources and formulated in the user's chosen natural language.

7 Conclusion

This paper has described a general model of human-computer communication in natural language, called Database Semantics, and shown applications to the www such as the improvement of recall and precision, inferencing, and a user-friendly processing of answers. In the talk, the current Java implementation of DBS will be demoed in the speaker- and the hearer-mode.

References

Abney, S. (1991) "Parsing by Chunks," in R.Berwick, S. Abney, and Carol Tenny (eds.) *Principle-Based Parsing*, Kluwer Academic Publishers.

Berners-Lee, T., J. Hendler,& O. Lassila (2001) "The Semantic Web," *Scientific American*, 284(5).

Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283-308, Elsevier. (abbreviated as TCS'92)

Hausser, R. (ed.) (1996) *Linguistische Verifikation. Dokumentation zur Ersten Morpholympics*, Max Niemeyer Verlag, Tübingen.

Hausser, R. (1999/2001) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language*, Springer-Verlag, Berlin, New York, 2nd Ed. (abreviated as FoCL'99)

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

Hausser, Roland (2005) "Turn Taking in Database Semantics," to appear in H. Kangassalo et al. (eds.), Information Modeling and Knowledge Bases XVI, IOS Press Ohmsha, Amsterdam, 2005. Available at http://www.linguistik.uni-erlangen.de

Kycia, A. (2004) *Implementierung der Datenbanksemantik in Java*, MA-thesis, Universität Erlangen-Nürnberg.

Vergne, J. (2002). "Une méthode pour l'analyse descendante et calculatoire de corpus multilingues : application au calcul des relations sujet-verbe," *Actes de TALN* 2002, 63-74.