

# Representing and Reasoning in Computer-Readable Field Linguistics Archives

*Richmond H. Thomason*  
AI Laboratory  
University of Michigan  
Ann Arbor, MI 48109-2210  
U.S.A.

`rich@thomason.org`  
`http://www.eecs.umich.edu/~rthomaso/`

December 4, 2000

## Workshop on Web-Based Language Documentation and Description

Dec 15, 2000

Institute for Research in Cognitive Science (IRCS)  
University of Pennsylvania  
Philadelphia, Pennsylvania, USA

### Abstract

Looking at retrieval in field linguistics as a knowledge representation and reasoning problem, I explore in this paper the possibility of using a knowledge representation system known as CLASP to provide knowledge-based retrieval services in this domain.

## 1. Apology

All that I will try to do in the present paper is to present a certain point of view and to explain how I think that some ideas that have been developed in Artificial Intelligence applications might be helpful in providing computational support for field linguistics. The usefulness of the ideas can't be demonstrated without development and testing. I had hoped to have at least some preliminary results by now, but there were too many technical problems and too little time for me to reach this stage. Nevertheless, I hope that it will be useful to describe the program.

## 2. Field linguistics as a KRR domain

It is often helpful to separate systems that have to perform complex reasoning tasks into a module that stores representations of declarative information and other modules that manipulate this information. For instance:

- (1) One module might store information about a device that can malfunction—say a laser printer.
- (2) Another module might match observations against representations of general types of situations.
- (3) Still another, procedural module would recommend repair measures by matching observations to general concepts in the antecedents of rules.

Knowledge Representation and Reasoning (KRR is the area of Artificial Intelligence that deals (among other things) with these declarative, general representations, and with how to relate them to sound, efficient reasoning procedures. For instance, KRR is concerned with the representations and procedures in the ones that in our example would match an observed situation to a general concept.

What happens if we think about computational field linguistics as a KRR problem? We could try to design ways to represent general information about linguistic structures that will support the sort of reasoning that is involved in linguistic retrieval and hypothesis testing.

Of course, the materials that are acquired in the course of field linguistics have multiple uses. Here, I want to concentrate on the scientific use of these materials, as primary evidence for testing and suggesting linguistic hypotheses. And I will concentrate on text-formatted materials. Because linguistic hypotheses are often framed using theoretical concepts, and the relations of these concepts to observed patterns in the surface forms of a language is frequently very complex, it is often difficult for linguists to formulate a search that will reliably test a high-level hypothesis. The evidence that I have at the moment (all of it anecdotal, unsystematic, and unquantified) indicates that tools like regular expression search can help, but in many cases they provide only a crude approximation with very low precision and recall. In some cases, it seems to be impossible to formulate a search that will be helpful at all. In others (where you can search with low precision but good recall) you can narrow down the material that needs to be searched by hand, but a great deal of unaided searching remains to be done.

From a KRR standpoint, the problem is to design representations of general linguistic knowledge that would support testing of high-level linguistic hypotheses using corpora such as lexicons and texts.<sup>1</sup>

Naturally, it is linguists specialize in the design of representations of general linguistic knowledge. But with retrieval in mind, we may need to consider modifications of linguistic formalisms, hoping these will be minimal. Usually, linguistic representations were formulated without any special reasoning purpose in mind, and certainly without attention to the needs of the reasoning on which I propose to concentrate. Here, we can learn from the experience of computational linguistics; to support linguistic reasoning such as parsing and speech recognition, it may be useful to rethink the representations that are used in linguistic theory, though it is important to retain a format that is linguistically natural and intelligible to linguists.<sup>2</sup>

### 3. Taxonomic logics

Two elements figure in the abstraction that separates theoretical linguistic concepts from more observable ones: *taxonomic generalization* and *roles*.<sup>3</sup> Consider, for instance, the concept of the grammatical number of a noun. *Noun* is a general concept applicable to words. *Number of a noun* is also a general concept, but it is relational.

There is a family of knowledge representation systems that center on these representational devices: the *taxonomic logics*.<sup>4</sup>

All the systems in this family provide a language in which *concept definitions* can be supplied, and provide a mechanism for expressing relational definitions. For instance, in any of these systems you could define a Verb to be a Word whose syntactic category is V.<sup>5</sup> All taxonomic logics support the use of conjunction in concept definitions; they differ as to what other boolean connectives are allowed. Taxonomic logics also support the use of numerical quantifiers in definitions, so that you can define the concept, for instance, of a verb with at least one prefix and at least one suffix.

The fundamental reasoning service delivered by these systems is *automatic classification*, which arranges concepts into a *subsumption hierarchy* according to generality. Given any two concepts  $C_1$  and  $C_2$ , classification will tell us whether every instance of  $C_1$  must also be an instance of  $C_2$ —i.e., whether  $C_2$  subsumes  $C_1$ .

Concepts apply to individuals. But from the standpoint of a taxonomic logic, there is no difference between an individual and an *individual concept*—the conjunction of all the

---

<sup>1</sup>These materials, of course, will often be structured. We can expect a high level of markup in a lexicon, and probably have to be prepared for various levels of structuring in texts.

<sup>2</sup>One example is the use of feature structures (see, for instance, [9]). Another is the use of notation similar to the traditional one for phonological rules in connection with finite-state phonology (see, for instance, [7] and [1]).

<sup>3</sup>Combinations of these two elements have played an important part in the linguistic formulations that have been influenced by unification grammars. See, for instance, [4].

<sup>4</sup>Also known as *description logics*, *classification-based systems*, or *KL-ONE-style systems*. See [10] for an overview, historical background, and references.

<sup>5</sup>Not a very enlightening definition, but more complex and interesting definitions can be assembled. Some concepts, specifically marked as “primitive,” are undefined. Word would probably be treated as a primitive concept.

most specific concepts that apply to some individual. Classification enables us to perform recognition, by enabling us to compute, for any concept, whether it subsumes an individual concept. It also supports retrieval, assuming that we can index individual concepts to occurrences of expressions in a text.<sup>6</sup> We can think of a concept as a query, so that retrieval for a concept  $C_2$  is simply a matter of recovering the positions in a text that are occupied by an expression whose individual concept  $C_1$  is subsumed by  $C_2$ .

The CLASSIC system is a taxonomic logic developed at AT&T Laboratories. (See [8, 2, 3].) It is low in expressive power (i.e., it admits relatively few resources in the definitions it supports), but, in part because of its simplicity, it is reliable and well documented. In CLASSIC, the sample definition we gave of Verb would be formulated as follows.<sup>7</sup>

```
(cl-define-concept 'Verb' (AND Word
  (ALL syncat (ONE-OF V)))
)
```

#### 4. CLASP

For purposes of linguistic description, CLASSIC suffers from one glaring defect: it is unable to talk about subexpressions and their relative positions; that is, it is unable to talk about strings. But an extension of CLASSIC called CLASP, described in [5], supports definitions concerning string types, combining regular expressions with the concept hierarchies and roles of CLASSIC.

Devanbu and Litman envisaged an application of CLASP involving the specification of plan execution traces, consisting of sequences of actions. They illustrated this application with plans involving message processing switches. The following definitions, taken from [5], illustrate this intended use.

```
(DEFINE-PLAN (1)
  Plan
  (PRIMITIVE
    (AND Clasp-Thing
      (Exactly 1 initial)
      (All initial State)
      (EXACTLY 1 goal)
      (All Goal State)
      (Exactly 1 plan-expression)
      (ALL plan-expression
        (LOOP Action)
      )))
)
```

---

<sup>6</sup>This indexing is nontrivial. But we can expect it to be available if our corpus is a lexicon that is at all well structured (since a lexicon should provide complete information about its lexemes). The availability of indexing in other corpora would depend on the cost of marking the corpus appropriately. This would depend in turn on the availability of automatic markup tools and on the human resources we are willing to devote to the task.

<sup>7</sup>The construct ONE-OF creates a concept satisfied only by an enumerated list of items, so that (ONE-OF V) restricts the value of syncat to V.

This definition says that a plan is something that has a plan-expression, which in turn consists of a sequence of actions, as well as an initial state and a goal state. (The “plan-expression” of a plan is a representation of the sequence of components that make up the plan. “LOOP” is the Kleene \*. A primitive concept is not exhausted by its definition—that is, the definition provides a necessary but not a sufficient condition for the concept.)

```
(DEFINE-PLAN (2)
  Originate-and-Dial-Plan
  (AND
    (AND Plan
      (ALL plan-expression)
      (SEQUENCE
        (Caller-Off-Hook-Act)
        (Connect-Dialtone-Act)
        (Dial-Digits-Act)
      )))
  )))
```

This defines an `Originate-and-dial-plan` as a sequence of three actions, where each action has to satisfy an associated concept, given in the definition. This is not a primitive definition, so that any individual that satisfies the definition will be recognized as an `Originate-and-Dial-Plan`.

Although it was designed with plan traces in mind, CLASP is equally—perhaps more—suitable for linguistic applications. These applications involve a large number of concepts that can be defined in terms of relatively few primitives. As we know from the success of finite-state techniques in natural language processing, regular expressions provide a natural and relatively expressive mechanism for characterizing linguistic patterns. The representational techniques inherited from unification grammar and incorporated in a several successful linguistic theories (including HPSG and LFG) carry over with relatively little distortion to taxonomic logics, since the *attributes* (roles with exactly one filler, that is, relations that correspond to partial functions) of the taxonomic logic do not differ in any essential respect from the *features* of feature structures.

In the remainder of this paper, I will illustrate how these representational techniques could be used to organize information in one application with which I am especially familiar; the lexicon of Montana Salish developed by Sarah G. Thomason. I will briefly described how relevant morphological knowledge about Montana Salish could be encoded in the form of CLASP definitions, and then indicate how the retrieval mechanism sketched above could work in some realistic examples.

In this version of the paper, I am trying only to convey a convincing general picture of the overall idea. For these purposes, it seems more appropriate to work with informal definitions in English rather than with the fully formalized CLASP equivalents.

## 5. Formalizing some Montana Salish morphology in CLASP

I will assume that we are working with orthographic representations only, and that we have developed a concept hierarchy for symbols, including the concept of a Montana-Salish-symbol

(an orthographic symbol of Montana Salish). Since the orthography of Montana Salish is phonetic, we can define concepts like Labialized-Consonant and Uvular-Consonant on the orthography.

First, we define a Montana Salish *string* and a Montana Salish *word*.

A Montana-Salish-String has exactly one body. Its body is a sequence of Montana-Salish-Symbols. (3)

A Montana-Salish-Word is a Montana-Salish-String that has exactly one part-of-speech,<sup>8</sup> which is a Part-of-Speech. (4)

In the above definition, the notion of a part of speech is realized in two ways: as a concept, and as a role. This sort of duality is common in CLASSIC representations.

Second, we define and populate the various prefixes and suffixes of the language. Montana Salish has quite a rich inventory of affixes. To simplify things, I will only consider preposed subject and object particles and one type of locative prefix. As for suffixes, I will only consider lexical suffixes, the derived transitive suffix, and transitive suffixes. The definitions for affix concepts contain relatively little information. Positional information will be included in the definition of a verb. This is illustrated by the following definition of `Lexical-Suffix`.

A `Lexical-Suffix` is a Montana-Salish-String. (This is a primitive concept.) (5)

Third, we define and populate the Verb-Roots.

The definition of a verb is analogous to Definition (1) from [5].

A Verb is a Montana-Salish-String, satisfying the following conditions. (1) Its syntactic category is V. (2) Its body is a Montana-Salish-String, consisting of a string of preposed particles and prefixes, a Verb-Root and a string of suffixes. (3) It has exactly one verb-root, which is a Verb-Root. (6)

Note that this definition mentions a verb's root in two places: as a position in the body of the verb, and as a role filler.

To represent more refined information about the patterns that are found in verbs, we would need to defined a large number of more specialized verb concepts. The following is an example.

`Intrans_Subj-Locative-Lexical_Suf-Verb`. (7)

This is a Montana-Salish-Verb whose body is a sequence consisting of an Intransitive-Subject-Particle followed by a Locative-Prefix followed by a Verb Root followed by a Lexical-Suffix, and which satisfies the following conditions. (1) It has exactly one intransitive-subject-particle, which is an Intransitive-Subject-Particle. (2) It has exactly one locative-prefix, which is a Locative-Prefix. (3) It has exactly one lexical-suffix, which is a Lexical-Suffix.

## 6. Some retrieval examples

The following queries, obtained from Sarah G. Thomason, all correspond to real needs for information about the language.

**Case 1.** It would be useful to be able to check systematically for the cooccurrence patterns of locative prefixes and lexical suffixes. Here, we would want to retrieve Montana-Salish-Verbs having a locative-prefix and a lexical-suffix. The fact that we can represent the query as a CLASP concept ensures the feasibility of this retrieval.

**Case 2.** It would be useful to know if there are Locative-Prefixes that do not cooccur with certain Lexical-Suffixes. CLASP would not support the formulation of a quantificationally complex query like this. However, a procedure for retrieving combinations of Lexical-Prefixes and Lexical-Suffixes that are not realized in the lexicon could be written in LISP, the programming language that underlies both CLASSIC and CLASP.

This project is based on the hope that the retrieval needs of field linguists could be served by a system that only requires them to enter information in a way that is linguistically natural, and it would defeat this goal to require field linguists to become LISP programmers. We would therefore have to hope that a limited number of preassembled retrieval routines would cover most of the retrieval needs that arise. Whether this can be done would have to be established by testing and experimentation.

**Case 3 (A subcase of Case 1).** What Lexical-Suffixes does the Locative-Prefix  $\check{c}$  cooccur with? We can form the notion of the lexical-suffix of a Verb whose locative-prefix is  $\check{c}$ . CLASSIC supports the retrieval of all the individuals satisfying this condition.

**Case 4.** It would be useful to determine how valency is affected by certain combinations of prefixes and suffixes. This means that, for instance, it would be useful to search for verbs that have a locative-prefix, a lexical-suffix, and a derived-transitive-suffix. This could be done as a direct CLASP retrieval.

**Case 5.** It would be useful to find all words involving three or more successive consonants. This would be a straightforward CLASP query.

**Case 6.** To investigate reduplication, it would be useful to find words involving the pattern  $C_1 i C_1$ . This could be formulated as a regular expression, although the formulation is awkward. So, although CLASP could perform the retrieval, it doesn't allow the query to be formulated in a natural way. This suggests that it might be useful to design a retrieval language with patterns like  $C_1 i C_1$ , which could then be compiled into regular expressions.

## 7. Conclusion

Unfortunately, the project that is described here has not been implemented and tested. Given the time constraints that are responsible for this lack of progress, I can't hope realistically to initiate a testing phase of this project until I can manage to teach a seminar on the topic and

recruit a group of people who can work on the implementation and testing over an extended period. Until this testing is carried out, it is impossible to make confident claims for its workability. At this stage, I need to be prepared to find out that CLASP is not the best tool for this purpose. An approach based on typed feature structures, for instance, might prove to be more appropriate (see [4]).

The limited experience I have had with field linguists, however, does suggest that searches using general-purpose retrieval mechanisms, such as standard database format and retrievals, or regular expression matching alone, are not adequate for the purposes of linguistic research. I believe that an approach that is specially designed with the representation and retrieval needs of linguists in mind is likely to prove far more useful. Combining taxonomic reasoning with roles and regular expressions appears to be a very natural way to approach the design of such a system.

## 8. Acknowledgement

The help of Sarah G. Thomason and Lucy G. Thomason in providing data for this paper is gratefully acknowledged.

## 9. References

- [1] Evan L. Antworth. *PC-KIMMO: A Two-Level Processor for Morphological Analysis*. Summer Institute of Linguistics, Dallas, 1990.
- [2] Ronald J. Brachman, Alex Borgida, Deborah L. McGuinness, and Lori A. Resnik. The CLASSIC knowledge representation system, or, KL-ONE: The next generation. In N.S. Sridharan, editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA, 1989. Morgan Kaufmann.
- [3] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lori A. Resnik. Living with CLASSIC: When and how to use a KL-ONE-like language. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, San Mateo, California, 1991.
- [4] Bob Carpenter. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, England, 1991.
- [5] Premkumar Devanbu and Diane J. Litman. Taxonomic plan reasoning. *Artificial Intelligence*, 84(1–2):1–35, 1996.
- [6] Roger Evans and Gerald Gazdar. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216, 1996.
- [7] Kimmo Koskenniemi. Two-level model for morphological analysis. In Alan Bundy, editor, *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Los Altos, California, 1983. William Kaufmann, Inc.

- [8] Lori A. Resnik, Alex Borgida, Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, and Kevin C. Zalondek. CLASSIC description and reference manual for the common lisp implementation version 2.2. Technical report, AT&T Laboratories, 1993. (PostScript file for this document available by anonymous ftp from pogo.isp.pitt.edu. Location is /users/ftp/pub/classic-tutorial/uncompressed/manual.ps. This document is regularly updated.).
- [9] Stuart M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information, Stanford, California, 1986.
- [10] William A. Woods and James G. Schmolze. The KL-ONE family. In Fritz Lehmann, editor, *Semantic Networks in Artificial Intelligence*, pages 133–177. Pergamon Press, Oxford, 1992.