TOWARDS A GENERIC ARCHITECTURE FOR LEXICON MANAGEMENT

Cristina Vertan
Walther von Hahn,
University of Hamburg, Natural Language Systems Department
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
cri@nats.informatik.uni-hamburg.de, vhahn@nats.informatik.uni-hamburg.de

Abstract

In this paper we propose an architecture for a lexicon management tool MANAGELEX. This tool aims at a general environment for reading, updating and combining lexicons in different formats. The starting point is the already existing lexicon models MULTILEX and GENELEX. Each functionality (reading, updating and combining) is based on a corresponding model, which can be configured and maintained coherently.

1. INTRODUCTION

A large amount of lexical resources was developed during the last 15 years. Unfortunately, in the absence of a standard each application produced and used its own lexicon in a specific format and a specific model, according to particularities of language, system functionality and available physical resources. Reusable lexical resources, however, could noticeably reduce the cost of development of NLP applications. Moreover, during research projects, lexicon requirements may change over the run time of the project, and maintaining a suitable lexicon is expensive and time-intensive work.

The problem of standardization appeared as an absolutely and urgent necessity, and several projects were carried out in this sense (v.Hahn 2000). The task is quite difficult because it implies at least two components : standardization of the format and standardization of the model. Moreover, these two components are not completely independent. For the former it is general agreed today, that the starting point is a SGML –based format. Several SGML-lexicon standard formats were already proposed (EAGLES, OLIF, SALT) (Lieske & al. 2001, Melby 1999). It is, however, necessary that we have not only a standard set of tags but also a standard model of a lexicon representation. As a result of this insights, several projects tried to develop a standard and general model for lexicons. The most well-known formalisms after this phase are GeneLex and Multilex.

2. STANDARD LEXICON MODELS. STATE OF ART

Although having many architectural features in common, Genelex is abstracted basically from a French monolingual lexical model, whereas the Multilex architecture is genuinely designed as a multilingual language-independent general structure, trying to include all language specific models (EAGLES 1996). At least, as quoted in one of the final reports (Praprotté & al. 1993), Multilex *"is based on a consideration of the following languages: English, German, French, Spanish and Italian, and to lesser degrees Dutch and Greek".* Compared to the multitude of (at least) European languages we observe that the Slavonic family was not taken into consideration, and also a lot of other languages which bring in new linguistic features (for example Romanian, although it belongs to the Latin languages, it has several important characteristics, due to the Slavonic influence).

The MULTILEX architecture presented a generic model for a lexical entry, which can be used as a starting point for further developments. However MULTILEX, as other similar projects *"imposes constraints on the linguistic level. Each of these projects imposes its own notion of 'lexical unit' (lemma, word-sense, concept) and its own logical structure (Typed Feature Structures, Entity-relationship model, automata, trees,...)"* (Sérasset 1996).

With these constraints, a user at the moment cannot use the same system to manipulate two lexicons coming from different places. Some steps in this direction were done in MULTILEX, which originally proposed the development of tools to convert lexicons into MULTILEX format. The proposal was not further developed because, quoting the same final report (Praprotté & al. 1993) *"copyright problems, problems in converting and correcting dictionary data, a lack of consistency in the data"* made this proposal unreachable.

Much lexical work from completed projects cannot be used in follow-up projects because of one of the following reasons:

- The lexicons were produced with the help of systems that are not any longer maintained; thus nobody can provide an export facility.
- In some cases, lexicon definitions contain procedural elements, which cannot be used without the hosting system,
- Lexicons may contain too rich features, which are too expensive to remove from the files.
- Experimental lexicons may be inconsistent or contain entries with different granularity,
- Lexicons may be stored in a data base, whereas others are plain files and the export formats do not match,
- Lexicons differ in their linguistic classes, i.e., there is a more-to-more mapping between feature classes.

From another point of view the use of a specific format (for example MULTILEX) means to adapt a posteriori other systems' processes to read and work which such external formats. This is usually quite cost-expensive.

The situation is much more critical for small languages, and languages from Central and East Europe, for which lexical resources were developed quite ad hoc as they were needed for a certain project.

Although a lot of resources after a few years may be linguistically and technically outdated, about 60% of a dictionary with approx. 80 000 entries comprises the lexical core of very high and rather high frequency words, which remain stable in their syntactic and semantic properties over a long period of time. The other part (especially terminology) from time to time must undergo revision, updating or even replacements.

3. MANAGELEX A GENERIC LEXICON MANAGEMENT MODEL

Following the above considerations, we assume that for a rather long time from now, NLP applications will still have to deal with manipulations of non-standard lexical resources.

However, this is only possible with rather general lexical management tools for acquisition, comparison, manipulation and validation of lexicons, based on several abstract models.

In this section we propose a new architecture for a lexicon management tool (MANAGELEX), a tool, which is able to read, convert and combine lexicons, independent of their format, language or system requirements.
The general architecture of such a system includes (as shown in figure 1) 3 levels of abstraction (which follow the ANSI(1999) data modeling specifications): the meta model level, the model level and the real world level.

- The real world level identifies real (present), distinct objects, their concrete features, and the actual relation among them. In figure 1 this corresponds to the encoded lexicons (DocA, DocB) and their structure (StructA, StructB)

- The model level groups real world objects and present features into object and attribute classes and recognizes possible relationships among object classes. On this level our architecture has 3 tools:
  - A tool for reading and updating a lexicon (acquisition and editing tool),
  - a tool for encoding and decoding (encoding / decoding tool) and
  - a tool for mapping two lexicons, possibly with different structure (mapping tool)

- The meta model level, classifies types of elements appearing on the model level and the abstract relations among them, situation independent. Accordingly, we propose

  - A generic lexicon model (LexMod) which provides a rather rich model of possible lexical information. Here, every linguistic feature, with their possible values which may occur in a set of languages (at least European) are specified (MULTILEX together with the MILE (Calzolari & al. 2001) model (defined in the frame of the ISLE project) are a good starting point). A flexible formal specification will be provided for this model. The model will also allow for new categories, joining as well as splitting of existing categories.

  - A generic encoding model (Encod), which specifies the way of combining the linguistic information in a specific entry and lexicon structure. The model should also include options for encoding files in the new generally agreed SGML-standards as OLIF or SALT (Lieske & al. 2001; Melby 1999).

  - A mapping model (MAP), that specifies modalities of combining two lexicons and takes into account problems like mutual gaps and complex categories.

Given this architecture, we now explain the functionality of the envisaged system in three situations:

1. Building / updating a lexicon.

Input: Lexicon definition from LexMod, Encoding Model Encod,
Output: Lexicon interface, lexicon file

The operation is mainly performed by the acquisition/editing tool. The interface of this tool is built automatically according to the characteristics selected from LexMod for this particular lexicon. The output of this tool is a data structure recording the structure of the lexicon LexA. The encoding / Decoding Tool uses this data structure and the Encoding module and produces and encoded lexicon DocA.

2. Reading a lexicon.
Input: Lexicon file, Encoding Model Encod,
Output: -

This operation requires first the identification of the encoding and the generation of the corresponding linguistic structure (StructB). Responsible for all these is the encoding tool

3. Join of two lexicons (LexA and LexB)
Input: General Lexicon definitions from LexMod, lexicon definitions from StructA and StructB, mapping models MAP
Mapping models MAP
Output: Lexicon file

This is the most challenging operation. The mapping tool has to use not only the structure of the two lexicons (StructA and StructB) and the mapping model (MAP) but also the generic lexicon model (LexMod). This is required for example in case of different names for the same linguistic feature. The resulting structure contains data consistent with both lexicons. Furthermore a new lexicon can be encoded as described above.

4. CONCLUSIONS

In this paper we described a model of a possible lexicon management tool, which can deal with frequent problems in lexicon acquisition / maintenance. The presented architecture is still in prototyping phase. We envisage to develop it in the frame of an European project. How ever for the moment we will take into account the European languages. Extensions to other language should be possible one the system reaches a stable version. The system is not intended to replace the actual already defined standards, but to supply the use and reuse of the already developed non-standard lexical resources

REFERENCES

ANSI-American National Standard Institute(1999), Standard X3. 138-1988, *Information Resource Dictionary System (IRDS)*

Calzolari, N. and A. Lenci and A. Zampolli and N. Bel and M. Villegas M. and G. Thurmair G., (2001) "The ISLE in the Ocean – standards for Multilingual Lexicons (with an Eye to Machine Translation*)", Proceedings of MT Summit VIII, Santiago de Compostella, 2001*

EAGLES (1996) "Input to the EAGLES architecture work: survey of MULTILEX", *http://www.ilc.pi.cnr.it/EAGLES96/lexarch/node4.html*

v.Hahn, W (2000), "Standards in Natural Language Processing – New Steps in Language Engineering", in *Standards in Information technology S. Nedevschi and K. Pusztai (Eds.*), Casa Cartii de Stiinta, Cluj.

v.Hahn, W. (1999), "Metamodelling of Lexical Acquisition Tools"*, Proceedings of EUROLAN '99*, Iasi.

Lieske Ch. and S. McCormick and G. Thurmair (2001), "The Open Lexicon Interchange Format (OLIF) comes of Age", *Proceedings of MT Summit VIII, Santiago de Compostella*

Melby, A. K. (1999), "SALT: Standards-based Access service to multilingual Lexicons and Terminologies", *http:// www.ttt.org*

Paprotté, W. and F. Schumacher(1993), "MULTILEX – final Report WP 9: MLEXd"*, Report MWP 8 – MS*

Sérasset G.(1996), "Recent Trends of Electronic Dictionary. Research and Development in Europe*", Report GETA-IMAG, CNRS, Grenoble*,

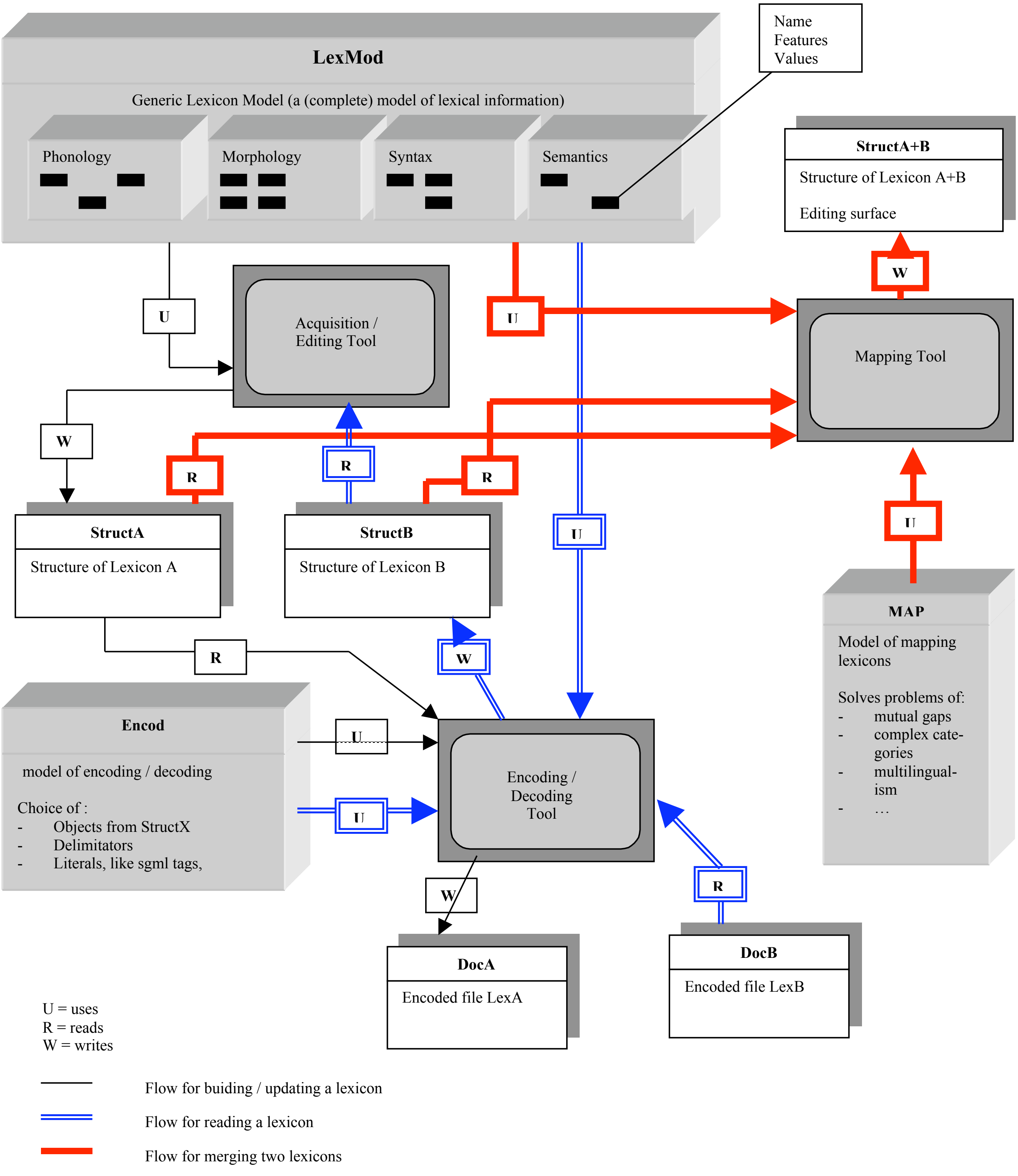

Figure 1: MANAGELEX: components and Workflow