XNLRDF, an Open Source Natural Language Resource Description Framework

Oliver Streiter
Institute of Western Languages and Literature,
National University of Kaohsiung
No. 700 National University of Kaohsiung
Road,
Kaohsiung, Taiwan
ostreiter@nuk.edu.tw

Mathias Stuflesser
Institute of Applied Linguistics,
European Academy of Bolzano
Viale Druso 1.
39100 Bolzano Bozen
mstuflesser@eurac.edu

Abstract

XNLRDF represents an unseen attempt to collect, formalize and formally describe language resources on a large scale so that they can be used automatically by computer applications. XNLRDF is intended to become a free software distributed in XML-RDF. This software is designed to be accessed by computer applications like Web-browsers, mail-tools, Web-crawlers, information retrieval (IR) systems or Computer Assisted Language Learning (CALL) systems. It proposes to replace idiosyncratic ad-hoc solutions for Natural Language Processing (NLP) tasks by a standard interface to XNLRDF. The linguistic information in XNLRDF covers a wide range of written languages and extends the information offered by Unicode so that basic NLP tasks like language recognition, tokenization, stemming, tagging, term-extraction etc can be performed. With more than 1.000 languages used in the Internet and their number continually rising, the design and development of such a software becomes a pressing need. In this paper we introduce the basic design of XNLRDF, the type of information the first prototypes will provide and describe the current state of the project.

1.XNLRDF as a Natural Extension of Unicode

1.1.Advantages of Unicode

With the advancement of Unicode, the processing of many languages, for which previously specific techniques were required, has become simplified. Unicode describes characters of language scripts by giving them a unique code point and properties like uppercase, lowercase, decimal digit, mark, punctuation, hyphen, separator or the script. Operations on the characters such as uppercasing, lowercasing and sorting are defined as well. Any computer application which is not endowed with particular linguistic knowledge is thus better off when processing texts in Unicode than in traditional encodings such as latin1, big5 or koi-r. With Unicode, the recognition of words, numbers and sentences may be performed without additional resources for many languages. Accordingly, various libraries for programming languages like Java, C++ or C have been developed to grant the programmer easy access to the information contained in Unicode.

1.2.Information needs beyond Unicode

The power of Unicode however is limited to characters and scripts only. Important notions such as language or writing system have no place in Unicode. For Unicode, two languages using the same script (Latin, Arabic, Hebrew etc) are essentially the same. A computer application operating on text documents, be it a Web-browser, a Web-crawler, an Information Retrieval System (IR), a Word processor or a CALL system thus have to integrate additional linguistic knowledge which should relate to the language, the region and time of text production, the writing standard, the orthography and the script. It is thus obvious that application processing text documents require, or at least profit from, additional information not provided by Unicode and which cannot be linked to Unicode.

First, an application might need to identify the encoding (KOI-R), the script (Cyrillic), the language (Russian), the standard and orthography, which sometimes are difficult to separate, ie. civil script, spelling before/after 1917 or 1956, of a document. The script, the language, the standard and the orthography create the writing system. Then, the application can, starting from the identified writing system, retrieve additional information which allows segmentation, stemming, hyphenation etc. A systematic approach will make services or tools available for small and unprivileged languages for which these services or tools remain otherwise unaccessible. It will provide a standard interface for the integration of linguistic information for those languages where resources already exist.

A web-crawler, for example, has to identify the encoding, the script (using Unicode) and the language of a document (using n-gram similarity) if these data are not specified in the meta-data of the document. In case of doubt, the crawler might fall back on the extension of the URL (e.g. xxx.xxx.nl) to restrict the range of possible languages (e.g. Dutch, Frisian, English) and to apply default assumptions (e.g. Dutch). As for the spelling, the system might make some additional (reasonable) assumptions, eg that the document complies with the most recent spelling standard. For a language like German, this however is actually quite tricky where about 10 standards are used. All these topics are not covered by Unicode. Once the encoding, script, language and orthography have been identified, text units (words, phrases) which are suitable for indexing are to be determined. In most cases the document will be segmented into words using a limited number of word-separating characters (e.g. empty space, comma, hyphen etc). For languages which do not mark word boundaries (e.g. Chinese, Japanese), the Web-crawler should index either each character individually (this is what Google does) or identifying words through wordlists and or rules. Spelling variants (humour, humor), writing variants (Häuser, Haeuser, H&aauml;user or 灣 , 湾 ,Wan), inflected word forms (come, came), abbreviations (European Union, EU) should be mapped onto their base forms to improve the quality of document retrieval. In order for the query to match documents, the search string has to be processed in the same way, applying language identification, script identification and segmentation.

Some of the required knowledge, eg about languages, regions or legacy encodings have been integrated into Unicode/Internationalization programming libraries for a limited number of languages in an ideosyncratic format, e.g. ICU, Basis Technology Rosette, Lextek Language Identifier etc. (for a detailed survey see http://www.unicode.org/onllinedat/products.html).

1.3.Difficulties to get information beyond Unicode

The need for a linguistic extension of Unicode has thus been long recognized and most of the information which applications as the one sketched above require is available in online resources. These applications, at least theoretically, could get them automatically from the Web. However, the resource or the information within cannot be accessed, extracted and integrated by these applications (and by humans only also with difficulties) for several reasons.

Difficulties to access the resources:

- Resources can not be found because meta-data are not available.
- The resource is not directly accessible for applications. (E.g. they require transactions like registering, submitting passwords, entering the credit card number, etc.).

Difficulties to extract the information:

- The resource is not formally structured.
- The information within the resource is formally structured but the syntax of the structure is not defined (as it could be through the usage of a DTD).

- The information is ambiguous, eg “*Abkhaz is a North West Caucasian language with about 105.000 speakers in Georgia, Turkey and Ukraine (...) The current Cyrrillic-based system*” (http://www.omniglot.com/writing/abkhaz.htm), leaving it actually unexpressed which region might or might not use the Cyrillic-based script.

Difficulties to process the information:

- The syntax is defined but the semantics of the units are not (as they could be through the usage of XML namespace etc.).
- The information in the different resources is not compatible, eg the “simple” notion of language varies greatly over resources. To give one example, what the Office of the High Commissioner for Human Rights (Universal Declaration of Human Rights) describes as Zapoteco (http://www.unhchr.ch/udhr/index.htm) is not covered in Omniglot (http://www.omniglot.com) and split into more than 50 languages by Ethnologue (http://www.enthnologue.com) and the Rosetta Project (http://www.rosettaproject.org).

2.Related Work and Available Resources

XNLRDF is embedded in a wide field of research activities which create, document and render accessible natural language resources. What makes XNLRDF particular in this field is its focus on Natural Language Processing resources on the one hand and the fully automatic access to the data by an application on the other hand. Nevertheless XNLRDF will try to profit from and comply to related projects and standards.

Repositories about the world's languages are available on-line. Among them figure Omniglot (http://www.omniglot.com), Ethnologue (http://www.ethnologue.com), The Rosetta Project (http://www.rosettaproject.org) and the Language Museum (http://www.language-museum.com). Although these resources offer rich information on scripts and languages, they are almost unusable for computer applications as they are designed to be used by human users. The difficulties to use Ethnologue, for example, derive from its focus on spoken languages and its tendency to introduce new languages where others just see regional variants of one and the same language. This problem is inherited in the Rosetta Project and the World Atlas of Language Structures (Haspelmath et al 2005). In addition, many sites use scanned images of characters, words and texts which of course are difficult to integrate.

OLAC, the Open Language Archives Community project (see http://www.language-archives.org/documents/overview.html), is promoting a network of interoperating repositories and services for housing and accessing NLP resources. Its aims and approaches are thus very close to those followed in XNLRDF and we foresee a considerable potential for synergy. However, the OLAC user scenario assumes a human user looking for resources and tools, whereas XNLRDF is designed to allow applications to find resources autonomously given a text document to be processed and a task to be achieved. Closely related to OLAC is the E-MELD project which supports the creation of persistent and reusable language resources. In addition, queries over disparate resources are envisaged (http://emeld.org).

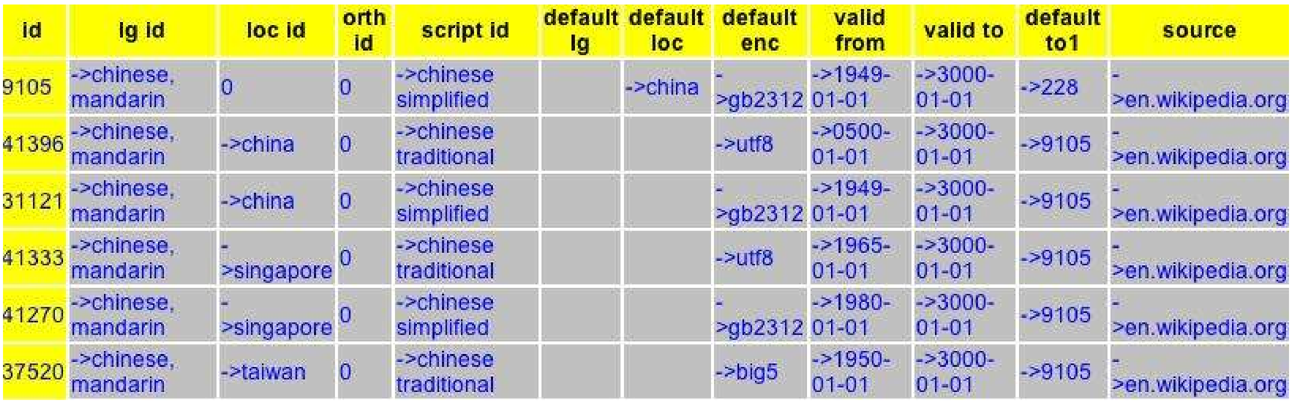
Data consortia like ELRA or LDC do not offer resources freely accessible, although machine readable meta descriptions are available through OLAC. Commercial transactions are required between the identification of the resource and the compilation into the application.

Project Gutenberg provides structured access to its 16.000 documents in about 30 languages through a XML-RDF. Unfortunately, information characterizing text T1 as translation of T2 is still not provided, that is although parallel corpora are implicitly present, they are not identifiable a such. Free monolingual and parallel corpora are available at a great number of sites, most prominently at http://www.unhchr.ch/udhr/navigate/alpha.htm (Universal Declaration of Human Rights in 330 languages), http://www.translatum.gr/bible/download.htm (The Bible), The European Parliament Proceedings Parallel Corpus (http://people.csail.mit.edu/koehn/publications/europarl/) and others.

3.Conceptual Design of XNLRDF

In order to give a word-to-word translation, for example, a web-browser has to know where to find a dictionary, for what purpose it can be used and under which circumstances. With only one such resource, a special function within the web-browser might handle this (e.g. a number of FIREFOX add-ons do exactly this). But with hundreds of language resources, a more general approach is required which not only involves adequate resources but meta-data with an NLP-specific meta-data dictionary and meta-data syntax. Such meta-data characterize a resource with respect to language, regions, script, encoding and format. NLP-operations like tagging or meaning disambiguation for annotated reading have then to be defined recursively in the metadata syntax: in this way, a tagger can call a tokenizer if it can't perform tokenization itself.

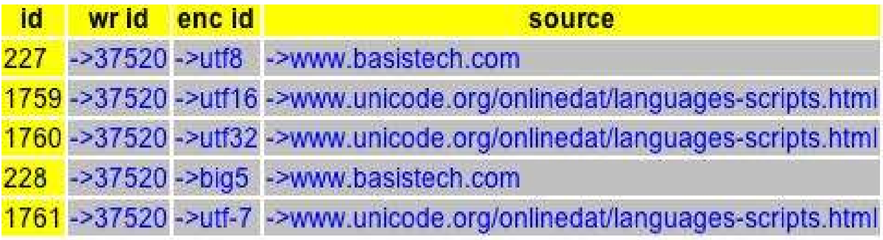
XNLRDF meta-data identify resources available within the XNLRDF data or outside XNLRDF. Central to the XNLRDF meta-data is the WRITING_SYSTEM. The WRITING_SYSTEM has a function similar to SUBJECT.LANGUAGE in the OLAC-metadata (Simons and Bird 2001), defined as "*A language which the content of the resource describes or discusses*". A writing system in XNLRDF is defined by the quintuple of LANGUAGE, LOCALITY, SCRIPT, ORTHOGRAPHY and STANDARD. Missing parameters in the WRITING (e.g. about LOCALITY) are used to express more abstract writings, eg a super-regional/national writing. Such abstract writing systems are motivated by text-documents which rightly or wrongly assume to be valid over a wider region and time (international treaties, religious texts, using the writing as lingua franca etc). This is illustrated in Plate 1, where a supra-national Chinese WRITING, eg for usage in the United Nations is listed.The same goes for dialects, which are treated as languages, whenever documents of that variant are attested (e.g. Akan Akuapem, Akan Asante, Akan Fante).



id	lg id	loc id	orth id	script id	default lg	default loc	default enc	valid from	valid to	default to1	source
9105	->chinese, mandarin	0	0	->chinese simplified		->china	->gb2312	->1949-01-01	->3000-01-01	->228	->en.wikipedia.org
41396	->chinese, mandarin	->china	0	->chinese traditional			->utf8	->0500-01-01	->3000-01-01	->9105	->en.wikipedia.org
31121	->chinese, mandarin	->china	0	->chinese simplified			->gb2312	->1949-01-01	->3000-01-01	->9105	->en.wikipedia.org
41333	->chinese, mandarin	->singapore	0	->chinese traditional			->utf8	->1965-01-01	->3000-01-01	->9105	->en.wikipedia.org
41270	->chinese, mandarin	->singapore	0	->chinese simplified			->gb2312	->1980-01-01	->3000-01-01	->9105	->en.wikipedia.org
37520	->chinese, mandarin	->taiwan	0	->chinese traditional			->big5	->1950-01-01	->3000-01-01	->9105	->en.wikipedia.org

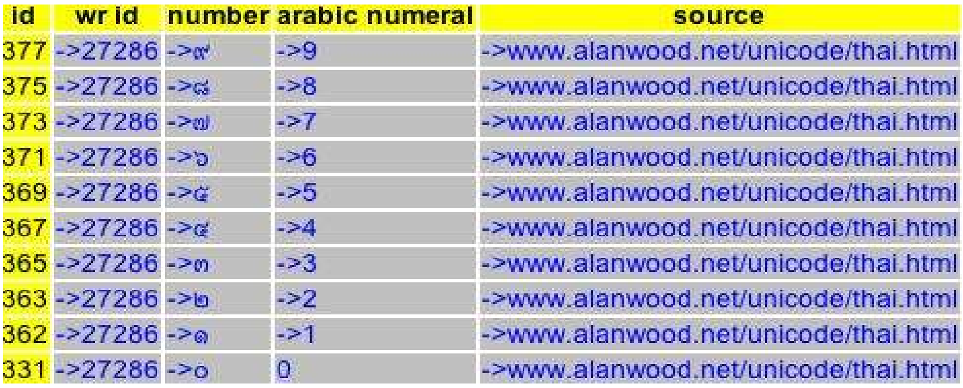
Plate 1: Chinese without locality as a super-regional language. In case of doubt, the application has to assume China as the locality where the text-document originated.

The so identified writing system is associated via a RESOURCE TYPE with the corresponding resources. WRITING_SYSTEMS stand in a many to many relation to ENCODING (Plate 2), NUMERALS (Plate 3), FUNCTION_WORDS (Plate 4). In addition to the RESOURCE TYPES shown here, WRITING SYSTEMS are related to CHARACTERS, SENTENCE_SEPARATORS, WORD_SEPARATORS, URLs (classified according to genres). In later versions, WRITING SYSTEMS will be related also to dictionaries, monolingual, parallel corpora and n-gram statistics.



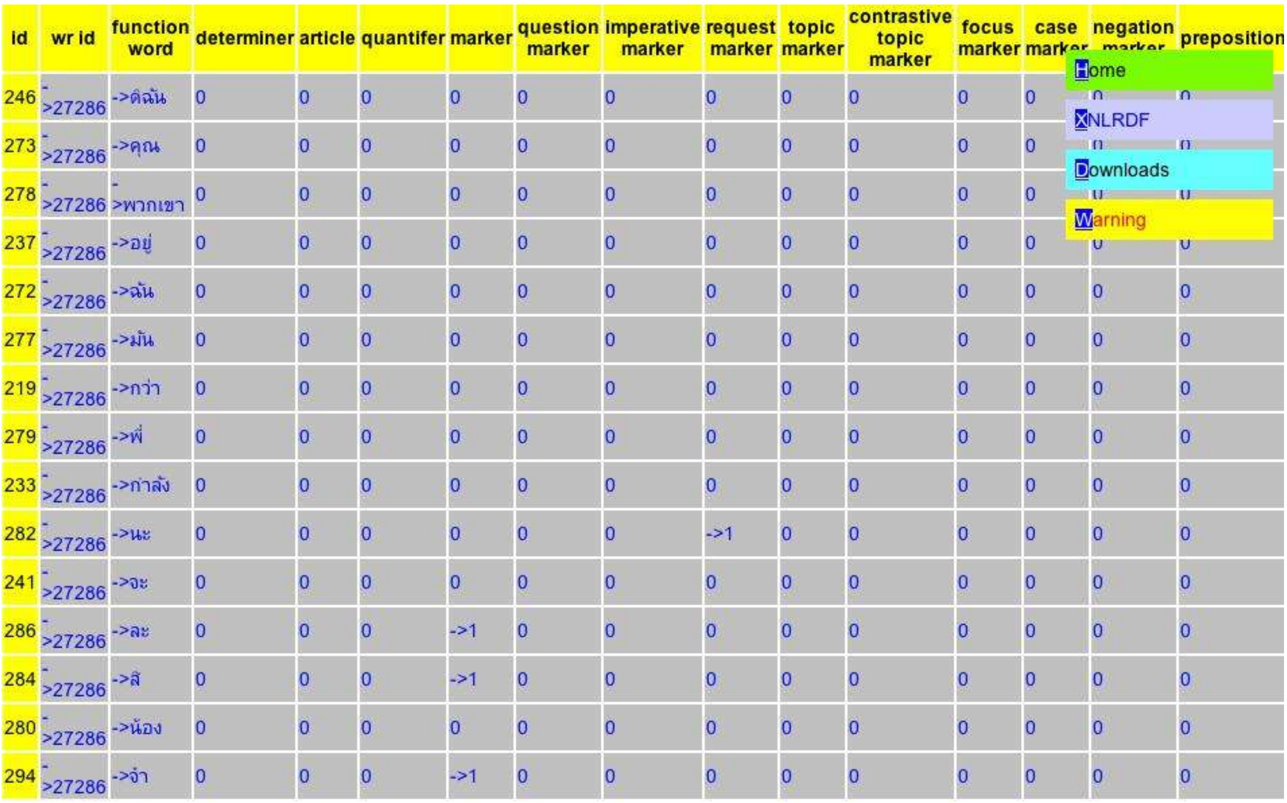
id	wr id	enc id	source
227	->37520	->utf8	->www.basistech.com
1759	->37520	->utf16	->www.unicode.org/onlinedat/languages-scripts.html
1760	->37520	->utf32	->www.unicode.org/onlinedat/languages-scripts.html
228	->37520	->big5	->www.basistech.com
1761	->37520	->utf-7	->www.unicode.org/onlinedat/languages-scripts.html

Plate 2: A writing system (Mandarin Chinese in Taiwan) related to ENCODING.



id	wr id	number	arabic numeral	source
377	->27286	->๙	->9	->www.alanwood.net/unicode/thai.html
375	->27286	->๘	->8	->www.alanwood.net/unicode/thai.html
373	->27286	->๗	->7	->www.alanwood.net/unicode/thai.html
371	->27286	->๖	->6	->www.alanwood.net/unicode/thai.html
369	->27286	->๕	->5	->www.alanwood.net/unicode/thai.html
367	->27286	->๔	->4	->www.alanwood.net/unicode/thai.html
365	->27286	->๓	->3	->www.alanwood.net/unicode/thai.html
363	->27286	->๒	->2	->www.alanwood.net/unicode/thai.html
362	->27286	->๑	->1	->www.alanwood.net/unicode/thai.html
331	->27286	->๐	0	->www.alanwood.net/unicode/thai.html

Plate 3: A writing system (Thai) related to NUMERALS.



id	wr id	function word	determiner	article	quantifer	marker	question marker	imperative marker	request marker	topic marker	contrastive topic marker	focus marker	case marker	negation marker	preposition
246	->27286	->ดิฉัน	0	0	0	0	0	0	0	0	0	0	0	0	0
273	->27286	->คุณ	0	0	0	0	0	0	0	0	0	0	0	0	0
278	->27286	->พวกเขา	0	0	0	0	0	0	0	0	0	0	0	0	0
237	->27286	->อยู่	0	0	0	0	0	0	0	0	0	0	0	0	0
272	->27286	->ฉัน	0	0	0	0	0	0	0	0	0	0	0	0	0
277	->27286	->มัน	0	0	0	0	0	0	0	0	0	0	0	0	0
219	->27286	->กว่า	0	0	0	0	0	0	0	0	0	0	0	0	0
279	->27286	->พี่	0	0	0	0	0	0	0	0	0	0	0	0	0
233	->27286	->กำลัง	0	0	0	0	0	0	0	0	0	0	0	0	0
282	->27286	->นะ	0	0	0	0	0	0	->1	0	0	0	0	0	0
241	->27286	->จะ	0	0	0	0	0	0	0	0	0	0	0	0	0
286	->27286	->ละ	0	0	0	->1	0	0	0	0	0	0	0	0	0
284	->27286	->สิ	0	0	0	->1	0	0	0	0	0	0	0	0	0
280	->27286	->น้อง	0	0	0	0	0	0	0	0	0	0	0	0	0
294	->27286	->จำ	0	0	0	->1	0	0	0	0	0	0	0	0	0

Plate 4: A writing system (Thai) related to FUNCTION_WORDS.

XNLRDF thus sets out to answer a computer application's questions like those listed below, where additional information might be added to the WRITING SYSTEM any time needed:

- *Where is language X spoken?*
- *Which languages are spoken in Y?*
- *Which script is used for language X in Y?*
- *What is the (default) encoding for language X in Y?*
- *How can words and sentences in X be identified?*
- *What are function words of X?*
- *How is stemming done for X?*
- *What are standard abbreviations in X?*
- *How can non-Arabic numbers be mapped onto Arabic numbers?*
- *Where are dictionaries/corpora related to X, what is their format and encoding?*
- *Where are parallel texts to language X in language Z?*
- *Which URLs can be accessed by a Web-crawler to gather monolingual corpora for a writing system, how are these Web-pages encoded and to what genre/register do they relate?*

4.Method and Implementation

The data-model is implemented in a relational database, which provides all means to control the coherence of the data, create backups and have people from different parts of the world working on the same set of data. For applications working with relational databases, these data can be downloaded under the GNU Public License as database dump (PostgreSQL). An additional goal is to make XNLRDF available in XML-RDF. RDF, a framework for the description of resource has been designed by the W3C to describe resources with their necessary meta-data for applications rather than people (Manola & Miller 2004). Whereas in the relational database the defaulting mechanism is programmed as a procedure, in XML-RDF defaults are compiled out. In this way, the information in XNLRDF can be accessed through simple a look-up with a search string such as 'Thai', 'Thailand', 'Thai;Thailand' etc. For the purpose of this lookup, XNLRDF adheres to the following standards, of which the first 3 reveal to be problematic.

- ISO-639-1for the 2-letter encoding of languages
- ISO-639-2 for the 3-letter encoding of languages
- SIL-codes (ethnologue) Version 14 for the encoding of languages
- Unicode-naming of scripts
- ISO-3166-1 alpha-2 and alpha-3 for the encoding of localities (countries, regions, islands, ...)

ISO-639-1 covers very few languages only, ISO-639-2 assigs more than one code to one language while both ISO norms assign the same code to sets of languages. SIL-codes may change from version to versions (about every 4 years), they do not cover historic languages, artificial languages and languages, they consider a language group or which exists only as written standard.

The situation for the encoding of language will improve with ISO/DIS 639-3 (presumably adopted in 2006) as it will combine the respective advantages of the SIL-codes and the ISO-codes. Until then, applications will continue to use the RFC 3066 standard for HTTP headers, HTML metadata and in the XML *lang* attribute. 2- and 3-letter codes are interpreted as ISO-639-1 or ISO-639-2 respectively. ISO-639-1 can be mapped on ISO-639-3 and ISO-639-2 is identical to ISO-639-3, so that in the future only ISO-639-1 (transitional) and ISO-639-3 will be needed (for more information on this very recent development consult http://en.wikipedia.org/wiki/ISO_639-3, http://www.ietf.org/rfc/rfc3066.txt and http://www.ethnologue.com/codes/default.asp). SIL-codes then will become superfluous and languages which are not written, can be removed from XNLRDF.

5.Envisaged Usage and Impact

The proof of the concept of XNLRDF will consist in compiling XNLRDF into an Mozilla-compatible RDF and integrated into an experimental Mozilla module. Not only is Mozilla a base for a great number of very popular applications (e.g. Firefox, Thunderbird, Bugzilla, Netscape, Mozilla Browser, Mozilla e-mail), it also disposes of an RDF-Machine which can be accessed via JavaScript and XPConnect (Boswell et al. 2002). A minor test-application of XNLRDF in Mozilla might thus have a tremendous impact.

CALL systems are another research area which will certainly profit from XNLRDF. Currently many CALL modules are freely available and, to some extent, language independent (e.g. hot potatoes at http://web.uvic.ca/hrd/halfbaked/), but in practice, they are often suited for Western languages only, eg they require a blank to separate words. In addition, they could profit from linguistic knowledge about function words, inflection, synonyms, to make them useful for a wider range of languages and generate better exercises and provide better feedback. A first implementation of NLRDF has been integrated into Gymn@zilla, a CALL systems which currently handles about 20 languages, with new languages added on a regular basis (Streiter et al 2005).

Web-crawlers and IR systems are other candidates which will certainly profit from XNLRDF. While most IR may be tuned to one or a few languages, they generally lack the capacity to process a wide range of languages. The large amount of NLP-systems integrated in Google shows the importance of linguistic knowledge in IR.

To sum up, we not only hope to bring many more languages to text-document processing applications, but to do this in a standard format which can be easily processed by XML or XML-RDF enabled applications.

6.Status of the Project

The project is still an unfunded garage project. In the current project phase we defined the base and implemented the first model in a relational database. An interface to that database has been created to allow for new data to be entered via the WWW. While more than 1000 WRITING_SYSTEMS have been inserted with LANGUAGE, LOCALITY, SCRIPT and ENCODING we currently are adding SENTENCE_SEPARATORS and WORD_SEPARATORS to them, so that texts can be segmented at least. As a next step we will enrich the WRITING_SYSTEMS with URLs of, preferably, dynamic web-pages, parallel corpora (Declaration of Human Rights, Bible, etc), a word-list of at least 1000 lexemes and a word list of inflected words with their base forms.

In the meantime we hope to attract more researchers to collaborate in the project, to share their knowledge, experience and possibly data since the ultimate goal will require the collaboration of a wide range of researchers around the globe. Special attention will be given to prototypic integrations of XNLRDF into various applications for testing and highlighting the impact of XNLRDF.

7.References

Boswell, D., B. King, I. Oeschger, P. Collins and E. Murphy, 2002. *Creating Applications with Mozilla*. Sebastopol: O'Reilly.

Haspelmath, M., M. S. Dryer, D. Gil and B. Comrie, (eds.) 2005. *The World Atlas of Language Structures*. Oxford: Oxford University Press.

Manola, F. and E. Miller, (eds.) 2004. *RDF Primer*, W3C Recommendation 10 February 2004, URL: http://www.w3.org/TR/rdf-primer/.

Simons, G. and S. Bird, (eds.) 2001. *OLAC Metadata Set,* URL: http://www.language-archives.org/OLAC/olacms.html.

Streiter, O., J. Knapp, L. Voltmer, C. Vettori, M. Stuflesser and D. Zielinski, 2005. Dynamic Processing of Texts and Images for Contextualized Language Learning, *Proceedings of the 22nd International Conference on English Teaching and Learning in the Republic of China (ROC-TEFL)*, Taipei, June 4-5, pp 278-298.