A Model for Interoperability: XML Documents as an RDF Database

Gary F. Simons	**SIL International**	**gary_simons@sil.org**
Brian Fitzsimons	**University of Arizona**	**fitzsimo@u.arizona.edu**
D. Terence Langendoen	**University of Arizona**	**langendt@u.arizona.edu**
William D. Lewis	**California State University, Fresno**	**wlewis@csufresno.edu**
Scott O. Farrar	**University of Bremen**	**farrar@uni-bremen.de**
Alexis Lanham	**University of Arizona**	**alexis@email.arizona.edu**
Ruby Basham	**University of Arizona**	**rubyb@email.arizona.edu**
Hector Gonzalez	**California State University, Fresno**	**hexgonzo@csufresno.edu**

Abstract

We propose a model for a Resource Description Format (RDF) database for interlinear glossed text (IGT) created from documents encoded in the Extensible Markup Language (XML) using markup metaschemas. A metaschema, constructed using the Semantic Interpretation Language (SIL) (Simons 2004) maps XML-encoded documents to a common semantically rich RDF database. The RDF database in turn can be searched using RDF-search engines providing the key functionality of a database management system (DBMS). Simons et al. (2004) gives a proof of concept of the model by mapping differently encoded XML lexicons to a common RDF form. Search capability is provided across these data using SeRQL, a SQL-like query language built around the Sesame RDF database program. In this paper, we extend these results to corpora of interlinear glossed text obtained from various sources, including some from the Web following Lewis (2003), combined with a language profile for each language variety, which provides basic grammatical information about that variety.

1. Introduction

Since its inception, the EMELD project has stressed as one of the most basic tenets of best practice the systematic use of the Extensible Markup Language (XML) for the interchange and archiving of data on endangered languages.[1] This has led to the adoption of best-practice proposals from the linguistics community, for example that of Hughes, Bird and Bow (2003) (henceforth HBB) for the XML encoding of interlinear glossed text (IGT). By adopting these formats, the EMELD project is well on its way to achieving its best-practice goals; but even with these formats in place and widespread use, much of the information contained in the markup is not comparable across documents. Even within a document, certain information may be misinterpreted or uninterpretable by users other than the document's creator. For preservation, this is undesirable, since losing the meaning of markup is nearly as bad as losing the data itself.

[1] For presentation at the EMELD Workshop on Databases, Detroit MI, 2004-07-15. Because of the experimental nature of the approach we have taken, this paper is incomplete in several respects. We hope to be able to have a more complete version for distribution at the workshop.

Furthermore, time can render markup conventions opaque that seem perfectly obvious when the analysis is first done.

Consequently the EMELD project has stressed the need for the creation of resources that define the space of *semantic* possibilities for linguistic markup, including interpretation of the structure of XML documents, interpretation of elements and attibute names, and interpretation of annotations appearing as content. It has also undertaken the creation of one such resource, the General Ontology for Linguistic Description (GOLD); see Farrar, Lewis and Langendoen (2002), Farrar and Langendoen (2003). In addition, with data in a semantically interoperable format – one in which the meaning of markup is explicit – cross-language, cross-theory search will be enabled.

Our challenge then is how to relate best-practice XML documents to semantic resources such as GOLD. To answer this, Simons (2003, 2004) proposes the use of *metaschemas* whereby best-practice documents are semi-automatically restructured and mapped to GOLD. A proof of concept for this approach using differently encoded XML lexicons for Hopi, Potawatomi and Sikaiana is given in Simons et al. (2004).

Assuming that the best-practice recommendations emerging from the EMELD project have made the data safely preserved, the question then arises as to the purposes the data might serve. Our work makes a step in the direction of bringing large amounts of IGT data out of hibernation, making them accessible for search and comparison. In particular we focus on how the Resource Description Framework (RDF) can be used to produce robust linguistic databases that (1) conform to the EMELD vision of best practice (2) have a distributed data model, (3) facilitate search, (4) and are scalable to large amounts of data. The paper is organized as follows: Section 2 gives a motivation for the RDF approach and presents the RDF database model. Section 3 provides a description of the IGT data that have been migrated to the RDF database. Section 4 describes how the metaschema is used in the migration process. And finally, Section 5 presents the utility of the resulting database by illustrating several search possibilities over the distributed database.

2. The Resource Description Framework

In the following section we discuss the benefits of using database technology in conjunction with linguistic data conforming to best-practice. We first present the motivation for the RDF approach in general, and then briefly review the basics of RDF. We then present our RDF database model for interlinear glossed text.

2.1. From XML to RDF

A *database* is an organized collection of data that is organized so as to facilitate querying and maintaining the information. Databases are often built using a Database Management System (DBMS), which not only provides the framework for defining structure and content of data, but also a facility to perform queries across the data. But because accessibility to DBMS systems can become increasingly opaque over time due to proprietary and non-standard encoding formats, the use of DBMS is not considered best practice for archiving and long-term preservation. In contrast, the use of XML has been shown to be a viable standard for the foreseeable future due to its universally accessible encoding format (Bird and Simons, 2003). Furthermore, XML is designed to be able to represent many aspects of database structure, and since it is a markup language, it is particularly suitable for representing the structure of linguistic data, in which the content is enclosed in (or associated with) elements that represent its structure. An example is

given in Figure 1, in which the string *akkan* occurring in an IGT document in HBB format (described in section 3.3) is analyzed as a word composed of two morphemes, the first of which consists of *akka*, categorized 'v' and glossed 'leave', and the second consists of *n*, categorized 'suff' and glossed 'RECENT_PAST_TENSE'.[2] Finally, since XML documents can be posted on the Web (and rendered human-readable by means of stylesheets), many different documents conforming to the same best-practice standard can be queried and searched as a distributed database. For certain kinds of data, notably IGT, already posted on the Web but not in a suitably structured format, tools can be developed to *migrate* those data to a best-practice XML encoding (Lewis 2003).

```
<word><item type="text">akkan</item>
<morphemes>
    <morph>
        <item type="text">akka</item>
        <item type="gloss">leave</item>
        <item type="pos">v</item></morph>
    <morph>
        <item type="text">n</item>
        <item type="gram">RECENT_PAST_TENSE</item>
        <item type="pos">suff</item></morph></morphemes></word>
```

Figure 1. XML representation of an elementary linguistic structure

The fragment in Figure 1 illustrates the need for semantic interpretation of even best-practice XML markup in order to make IGT resources fully interpretable and comparable. This need can be met in part by linking the encoded grammatical information to a resource like GOLD and the extragrammatical information (indicated here by the `type="gloss"` attribute) to a suitable "upper ontology" that provides definitions for common-sense concepts. But there is a more basic problem that results from the design of XML itself and how it is normally used.

XML provides two ways of relating objects to one another. If one object (character data or an element) is contained within another (an element), the relation can be indicated by enclosing the latter between the begin and end tags of the former. The other method is to point from one element to another, or to create a third element (representing the relation) that points to the related elements. XML structures that don't use pointers are linearizations of tree diagrams; accordingly Figure 1 can be graphed as the tree in Figure 2.

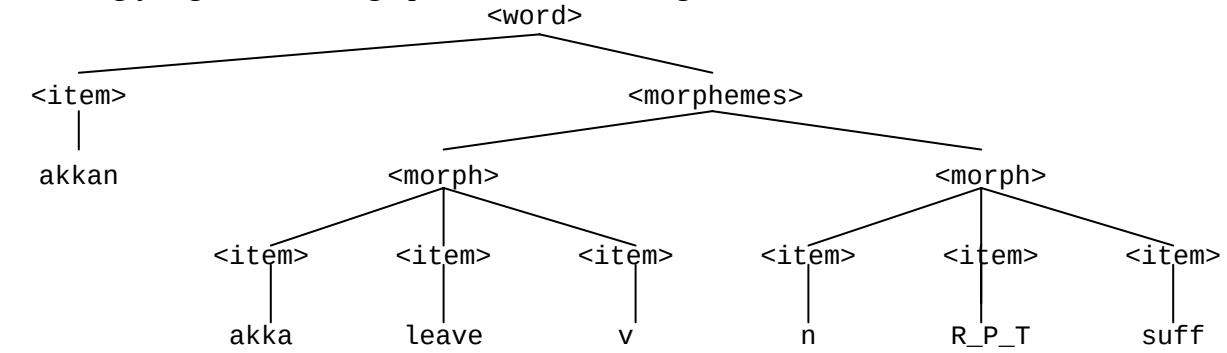

Figure 2. Tree structure of the XML object in Figure 1

[2] Note that the categorizations and glosses should be considered content; the structure is the bundling together of all the content pieces in Figure 1.

Since XML documents must contain elements but don't have to have pointers, the default structure of such a document is a tree. Consequently if a data structure can be represented by a tree (especially if the normal display format of that structure is interpretable as a tree), it will naturally be represented by a tree structure in XML, even if there is a better way of representing it using a different graphical structure. In particular, the HBB model of IGT is a tree structure which is derived from its typical display format in linguistic documents, by grouping together vertically aligned elements.[3] A better way of representing IGT is proposed in Simons (1997), in which recurring text elements (i.e. wordforms and morphemes) are represented only once, with attributes specifying where they occur in the text.

However, rather than recommend that HBB be replaced by a better best-practice XML schema using pointers, we propose to retain it, first because it is relatively straightforward (though not always easy) to migrate legacy IGT data in various formats to HBB, and second because we can extend Simons (2004) metaschema to map HBB documents to the desired structure in RDF,[4] which is designed to represent any desired graphical structure directly, without favoring tree structures.

In addition to recording the occurrence and grammatical properties of text segments within larger text structures, IGT can be and often is used to interpret the relations among those segments. For example, the word *akkan* in Figure 1 can be interpreted as a form of the verb stem *akka* inflected for recent past tense, with the suffix *n* expressing the inflection. In order to make such interpretation explicit, we require that IGT be accompanied by a *language profile* (LP), an XML document that at minimum specifies the lexical categories appearing in the IGT and the grammatical features that they are inflected for. A fragment of the Mutsun language profile based on the IGT we received appears in Figure 3. In this profile, we describe every lexical category of the morphemes appearing in the IGT file and identify the concept in GOLD closest in meaning to it, and the grammatical features for which it may be inflected. Then for each inflectional feature, we specify its possible values. Once the profiles are transformed to RDF along with the IGT files, it becomes possible to compare structural aspects of the data across the various languages, since the structural identifiers are all mapped to a common resource, and certain information that is implicit in the IGT is made explicit.

```
<POS abbrev="v">
    <label>Verb</label>
    <concept>gold:Verb</concept>
    <inflectedFor target="Aspect"/>
    <inflectedFor target="Modality"/>
    <inflectedFor target="Person"/>
    <inflectedFor target="Voice"/>
    <inflectedFor target="Tense"/>
</POS>
…
<feature abbrev="Tense">
    <label>Tense</label>
    <concept>gold:TenseAttribute</concept>
    <value abbrev="PAST">
        <concept>gold:AbsolutePastTense</concept>
```

[3] HBB is clearly the best way to represent IGT as a tree. The alternative of representing it as a sequence of lines as in an HTML table loses the information provided by the vertical alignment, as does the original XML encoding described below in section 3.4.

[4] We had hoped to be able to normalize the RDF representation of IGT derived from HBB along the lines of Simons (1997), but have not yet been able to do so.

```
            <label>Past</label>
        </value>
        <value abbrev="RECENT_PAST_TENSE">
            <concept>gold:RecentPastTense</concept>
            <label>Recent Past Tense</label>
        </value>
        <value abbrev="REMOTE_PAST_TENSE">
            <concept>gold:RemotePastTense</concept>
            <label>Remote Past Tense</label>
        </value>
    </feature>
```

Figure 3. Fragment of the Mutsun language profile

2.2. A brief look at RDF

The basic building block of an RDF graph is the object-attribute-value triple `<O, A, V>`, where O and V are Web resources and A is a binary predicate that relates O to V. For example, `<LinguisticUnit1, transcription, "ahki">` is an RDF triple that specifies that a particular LinguisticUnit is transcribed as the string *ahki*. RDF provides a more general data model and a more expressive basis for encoding linguistic data than XML by itself does; the entity-relation diagram in Figure 4 below is an illustration of what can be expressed in RDF. In particular, RDF elements come with a partially predefined semantics. For example the RDF element <rdf:Seq> is interpreted as a container whose elements are sequentially ordered. With the addition of RDF Schema (RDFS), a syntactic extension of RDF, an even tighter predefined semantics is available, such as the elements <rdfs:Class>, <rdfs:subClassOf>, and <rdfs:Property>; see Brickley and Guha (2004) for details of RDF and RDFS semantics.

RDF and RDFS were originally designed for use in the development of the Semantic Web (Berners-Lee 2001), and so are compatible with current web-ontology efforts such the creation of the Web Ontology Language (OWL) (McGuinness and van Harmelen, 2004); in fact they are the basis of OWL. When RDF resources are mapped to OWL-based ontologies such as GOLD, the development of intelligent software using the power of logical inference that operates over the data becomes possible, something not available for data simply encoded in XML. Moreover, once best-practice XML data are migrated to a common RDF format, the resulting RDF-encoded material constitutes a highly structured database, which provides a much better basis for search than does XML (Broekstra, Kampman and van Harmelen, 2002). Open standards are being developed for search over RDF structures, including the RDF Query Language (RQL) (Karvounarakis et al., 2000) and RDF Data Query Language (RDQL) (Seaborne, 2004). We are using SeRQL, which requires that the RDF triples be loaded into an open-source database program called Sesame (*Users Guide to Sesame*, 2004). In section 5, we present the details regarding search and show, in particular, that complex queries can be constructed for RDF that go beyond what ordinary XML query languages, such as XQuery (Chamberlin et al., 2001), can handle. Finally, ontologies provide a stable semantic resource for the interpretation of RDF elements, ensuring *enduring practice* inasmuch as the RDF structures can be maintained in parallel with compatibly-encoded ontologies.

2.3. The proposed RDF database model

The RDF database model sketched in the graph in Figure 4 is derived from the HBB IGT documents together with the LP documents by the process described below in section 4. The

relation in each RDF triple is indicated by a diamond; its domain is a box whose arrow points to that diamond, and its range is a box pointed to by the diamond. The labels on each node are concepts either defined in GOLD (e.g. *MorphosyntacticInformation*) or in RDF (e.g. *rdf:first*). For example, *meaning* is a relation whose domain is *LinguisticUnit* and whose range is *Semantic Sense*. resulting in triples of the form `<LinguisticUnit1, meaning, SemanticSense1>`. The root of the graph is the box node *DescribedVariety*, defined in the GOLD namespace as a language variety. In all of our examples, it is not distinguished from the *RecognizedLanguage* (identified by an SIL language code) of which it is a variety. The LP for each language variety defines of a *MorphosyntacticAttribute* class, with subclasses including *PartOfSpeech*, *Feature* and *Value* for that variety. Thus although Figure 4 shows the relations *inflectedFor* and *possibleValue* as holding between *MorphosyntacticAttribute* as both domain and range, in fact the domain of *inflectedFor* is the subclass *PartOfSpeech* and the range is the subclass *Feature*, whereas the domain of *possibleValue* is the subclass *Feature* and the range is the subclass *Value*.

The IGT on the other hand provides a *LinguisticUnit* class contained in a particular language. These are analyzed into their constituent parts (subclasses of *LinguisticUnits*) as identified by the XML hierarchy by a list-processing loop on *LinguisticUnit* passing through *MorphosyntacticInformation*. For example the XML element `<word>` is translated to an instance of *CompleteLexicalUnit*, a subclass of *LinguisticUnit*. Another list-processing loop identifies the individual grammatical properties which are part of the *MorphosyntacticInformation* associated with a particular *LinguisticUnit*, and assigns each of them a particular *MorphosyntacticAttribute*. Since these are the same *MorphosyntacticAttribute*s defined in the LP, it is possible to determine what status each of those properties has in the language. Moreover, since these *MorphosyntacticAttribute*s are associated with GOLD concepts, it is also possible to compare data across languages regardless of the abbreviations and terms used for those concepts.

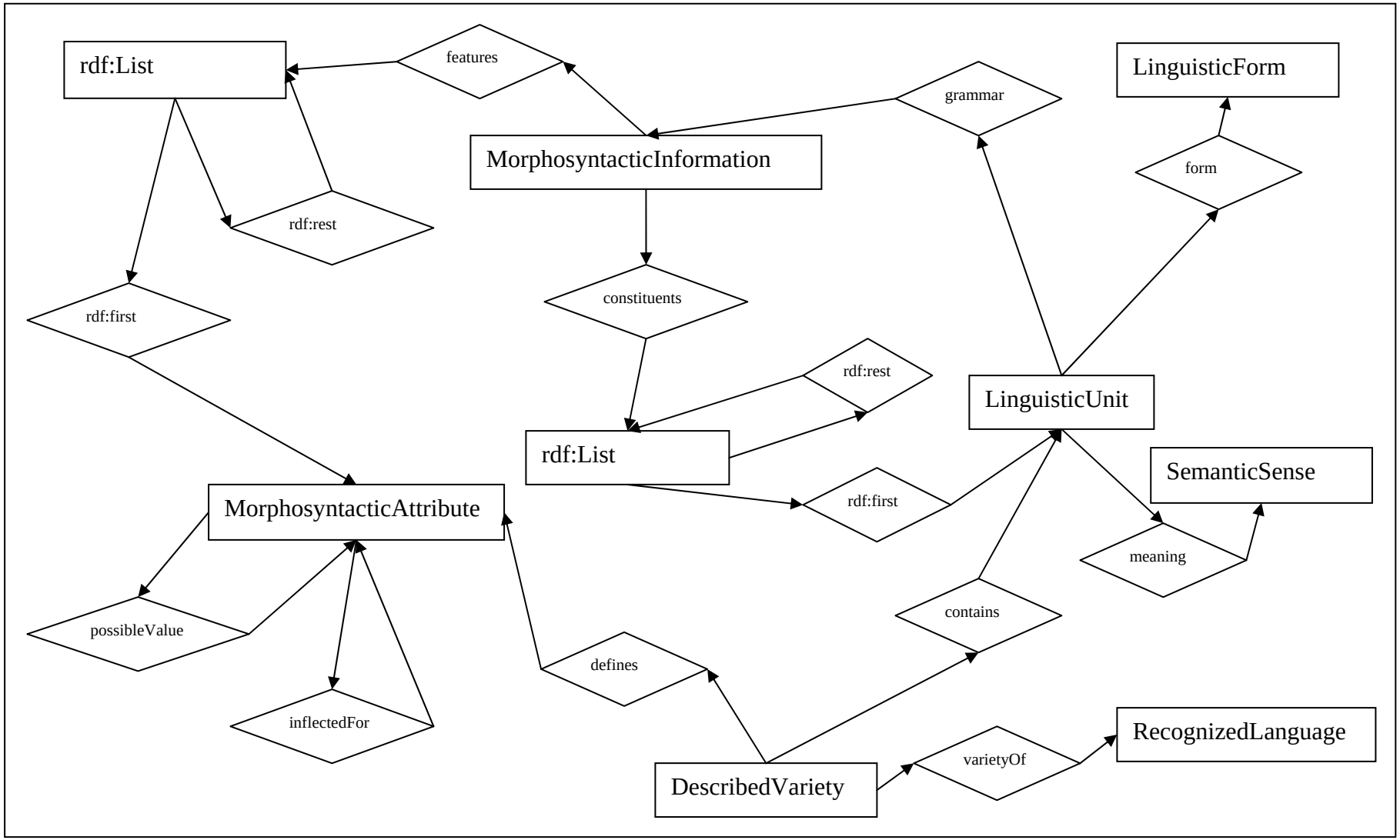

Figure 4. Entity-relation diagram (ERD) of RDF database of IGT

3. Profiles of the data sets used in the RDF migration

This section presents a brief profile of each data set used in the current project.

3.1. Archi

- SIL language code: ARC
- Source of data: Kibrik, Aleksandr E. (1998) Archi (Caucasian — Daghestanian). In *The Handbook of Morphology*, ed. by Andrew Spencer and Arnold M. Zwicky, pages 455-476. Oxford: Blackwell Publishers.
- size: 27 examples
- Original encoding: PDF file provided by the publisher; many of the examples appear in table on p. 471 and were converted directly to XML.
- Example of XML encoding: [http://emeld.douglass.arizona.edu:8080/Examples/GOLDMorphosyntaxExamples.xml]. Examples from many different languages appear in this file.

```
<phrase><item type="gloss">inside the apple</item>
<words>
<word><item type="text">aInš-l-a</item>
<morphemes>
    <morph>
        <item type="text">aInš</item>
        <item type="gloss">apple</item>
    </morph>
    <morph>
        <item type="text">l-a</item>
        <item type="gram">SG.INESS</item>
    </morph>
</morphemes>
</word>
</words>
</phrase>
```

3.2. Korean

- SIL language code: KKN
- Source of data: Han, Na-Rae (2003) *Morphologically Annotated Korean Text*. Linguistic Data Consortium, University of Pennsylvania, catalog # LDC2004T03 (distributed on CD-ROM).
- Size: 1,574 sentences; 41,024 words (with tokenized symbols); 77,173 morphemes; sentences were collected from Korean language newspapers
- Original encoding: Plain text, Korean transcribed in ksc-5601, a Hangul character set.
- Example of original encoding:

```
프랑스의 프랑스/NPR+의/PAN

르노 르노/NPR

자동차  자동차/NNC

…
^EOS(indicating the end of a sentence)
```

- Migration path: LDC provided software to convert the Korean text to Jordan (Yale) romanization. A Perl script was written and used to convert to HBB. We caught some transcription errors, which we sent back to the author.
- Example of XML encoding.

```
<phrase>
<words>
    <word><item type="text">phu.lang.su.nguy.</item>
    <morphemes>
        <morph>
            <item type="text">phu.lang.su.</item>
            <item type="gram">NPR</item></morph>
        <morph>
            <item type="text">nguy.</item>
            <item type="gram">PAN</item></morph></morphemes></word>
    <word><item type="text">lu.no.</item>
    <morphemes>
        <morph>
            <item type="text">lu.no.</item>
            <item type="gram">NPR</item></morph></morphemes></word>
...
</words></phrase>
```

3.3. Mutsun

- SIL language code: CSS
- Source of data: Natasha Warner and Lynnika Butler, University of Arizona, encoding of J. P. Harrington's Mutsun field notes
- Size: 5156 phrases, 12,548 words
- Original encoding: Shoebox
- Example of original encoding:

```
\id 0021
\idH 41/0154b
\idM
\ref Ar 1
\osA
\osH hÃ¡nnime â€™akkÃ¡n
\osM
\t hanni-me  akkan
\m hanni  -me  akka  -n
\g where? -you  leave -RECENT_PAST_TENSE
\p Q    -pron v  -suff
\f Where did you leave (it)?
\otA
\otH onde lo dejates?
\otM
\nt
\ns LB1 11/02, LB, NW 8/03
```

- Migration path: Converted to HBB ignoring certain fields in the original encoding.
- Example of XML encoding:

```
<phrase><item type="gloss">Where did you leave (it)?</item>
<words>
    <word><item type="text">hanni-me</item>
    <morphemes>
```

```
        <morph>
            <item type="text">hanni</item>
            <item type="gloss">where?</item>
            <item type="pos">Q</item></morph>
        <morph>
            <item type="text">me</item>
            <item type="gloss">you</item>
            <item type="pos">pron</item></morph></morphemes></word>
    <word><item type="text">akkan</item>
    <morphemes>
        <morph>
            <item type="text">akka</item>
            <item type="gloss">leave</item>
            <item type="pos">v</item></morph>
        <morph>
            <item type="text">n</item>
            <item type="gram">RECENT_PAST_TENSE</item>
            <item type="pos">suff</item></morph></morphemes></word>
</words></phrase>
```

3.4. Monguor

- SIL language code: MJG
- Source of data: Annotations of Arienne Dwyer's field recordings available at http://www.emeld.org/school/exhibits/monguor/av/emeld-sample-full.html
- Size: ~125 phrases (615 "events")
- Original encoding: XML (Note: lines of annotation are associated with time stamps)
- Example of original encoding

```
<layer l-id="MNADDA23Jan0301_23_28_o">
    <event e-id="495" start="0.0" end="2.455">Liukesangni bisa dasi zou
      hanyao, engbang sha?</event>
    <event e-id="496" start="2.455" end="5.928">Ni buraku shoujin bisa zou
      dasi zou bara bei engbang sha?</event>
</layer>
<layer l-id="MNADDA23Jan0301_23_28_pos">
    <event e-id="249" start="0.0" end="2.455">Vi-Nzr-ACC COPs-COND PNIpl
      Adv Vt Vt #COPo-PRTd</event>
    <event e-id="250" start="2.455" end="5.928">DET N Vi-COND Adv PNIPL
      Adv Vt HORT #COPo-PRTd</event>
</layer>
<layer l-id="MNADDA23Jan0301_23_28_dte">
    <event e-id="372" start="0.0" end="2.455">lost-?-ACC exist/enough-COND
      we again substitute send ok-PRTd</event>
    <event e-id="373" start="2.455" end="5.928">Those pillow.ends
      exist/enough-COND again# we again# offer HORT ok-PRTd</event>
 </layer>
 <layer l-id="MNADDA23Jan0301_23_28_te">
    <event e-id="3" start="0.0" end="2.455">If anybody is missing, could
      we send/offer it (#what?) in the future to substitute for
      this?</event>
    <event e-id="4" start="2.455" end="5.928">If the pillows ends are not
      enough, we can offer scarfs, right?</event>
 </layer>
 <layer l-id="MNADDA23Jan0301_23_28_tc">
    <event e-id="126" start="0.0" end="2.455">
```

```
        如果有漏掉的，我们再给你们补送，行不？</event>
    <event e-id="127" start="2.455" end="5.928">
        要是枕顶不够了，我们还可以送头巾，对不对？</event>
 </layer>
```

- Migration path: XSLT and Perl scripts (latter to convert delimiters to entities) to convert to HBB. Time stamps were removed, but they could have been retained.
- Example of HBB encoding (Note: codes in second phrasal gloss line refer to Chinese characters):

```
<phrase>
<item type="gloss">If anybody is missing, could we send/offer it (#what?)
     in the future to substitute for this?</item>
<item type="gloss">
     &#x5982;&#x679C;&#x6709;&#x6F0F;&#x6389;&#x7684;&#xFF0C;&#x6211;&#x4EE
     C;&#x518D;&#x7ED9;&#x4F60;&#x4EEC;&#x8865;&#x9001;&#xFF0C;&#x884C;&#x4
     E0D;&#xFF1F;</item>
<item type="text">Liukesangni bisa dasi zou hanyao, engbang sha?</item>
<words>
    <word>
    <morphemes>
        <morph>
            <item type="gram">Vi</item>
            <item type="gloss">lost</item></morph>
        <morph>
            <item type="gram">Nzr</item>
            <item type="gloss">?</item></morph>
        <morph>
            <item type="gram">ACC</item></morph></morphemes></word>
...
</words></phrase>
```

3.5. Hopi (1)

- SIL language code: HOP
- Source of data: Kenneth C. Hill (2003) Denominal and noun incorporating verbs in Hopi. In *Studies in Uto-Aztecan* (*MIT Working Papers in Endangered and Less Familiar Languages* 5), ed. by Luis M. Barragan and Jason D. Haugen, pp. 215-244. Cambridge, MA: MITWPL.
- Size: 244 examples
- Original encoding: Corel WordPerfect using WP InternationalA character set
- Migration path: Perl script used to convert to HBB. Data had to be massaged a bit by hand so the standard 3-line IGT format of examples used in text was not interrupted by extra returns; special characters were also identified and converted when the file was migrated to a UTF-8 encoding.
- Example of original encoding:

```
Puunat paayis teevep yooyoki.
recently three.times all.day be.raining
Recently it rained all day for three days.
```

- Example of HBB encoding:

```
<phrase><item type="gloss" xml:lang="en">Recently it rained all day for
     three days.</item>
<words>
    <word><item type="text">Puunat</item>
    <morphemes>
        <morph>
```

```
            <item type="text">Puunat</item>
            <item type="gloss">recently</item></morph></morphemes></word>
        <word><item type="text">paayis</item>
        <morphemes>
            <morph>
            <item type="text">paayis</item>
            <item type="gloss">three.times</item></morph></morphemes>
        </word>
    ...
    </words></phrase>
```

3.6. Hopi (2)

- SIL language code: HOP
- Source of data: Kenneth C. Hill, electronic version of *Hopi Dictionary Hopìikwa Lavàytutuveni: A Hopi-English Dictionary of the Third Mesa Dialect*, Tucson: University of Arizona Press, 1998. (Note: Example sentences appearing in the print dictionary are omitted from the electronic version we received permission to use.)
- Size: ~ 29,500 word entries including some that are not broken down into morphemes.
- Original encoding: Notebook I ver. 4.1, dated March 2, 1992 (flat database format with variable length text fields)
- Examples of original encoding:

```
 1 Headword........ |sá'akiw|ta
 2 Alphabetizer.... |sa'akiwta
 3 Inflected forms. |
 4 Combining forms. |
 5 Pausal.......... |
 6 Part of speech.. |vi.i.pl.
 7 Usage........... |
 8 Definition...... |be shouting, bursting out in shouts.
 9 Loanword source. |
10 Word breakdown.. |sá'a-k-iw-ta
11 Analysis........ |[make:noise(PL)-SGL-ST-DUR]
12 Underlying form. |/ sa'a -ku- -¡la -it¢a /
13 Cross-reference. |Cf. {sé'ekiwta,} fill it with noise (as when singing).
14 Semantic domain. |

 1 Headword........ |sá'akmanta
 2 Alphabetizer.... |sa'akmanta
 3 Inflected forms. |
 4 Combining forms. |
 5 Pausal.......... |
 6 Part of speech.. |vi.i.pl.
 7 Usage........... |
 8 Definition...... |be intermittently bursting out in shouts.
 9 Loanword source. |
10 Word breakdown.. |sá'a-k-man-ta
11 Analysis........ |[make:noise(PL)-SGL-POSTG-REP]
12 Underlying form. |/ sa'a -ku- -mán -t¢a /
13 Cross-reference. |
14 Semantic domain. |
```

- Migration path: Fields 1, 6, 8, 10 and 11, representing the morphosyntactic analysis of the dictionary entries, were converted to HBB format using a Java application by Gonzalez and Lewis (modified from that used in Simons et al 2004).
- Examples of HBB encoding:

```
<word>
<item type="text">sa'akiw|ta</item>
<item type="gloss">be shouting, bursting out in shouts.</item>
<item type="gram">vi.i.pl</item>
<morphemes>
    <morph>
        <item type="text">sá'a</item>
        <item type="gloss">make:noise(PL)</item>
    </morph>
    <morph>
        <item type="text">k</item>
        <item type="gram">SGL</item>
    </morph>
    <morph>
        <item type="text">iw</item>
        <item type="gram">ST</item>
    </morph>
    <morph>
        <item type="text">ta</item>
        <item type="gram">DUR</item>
    </morph>
</morphemes></word>

<word>
<item type="text">sa'akmanta</item>
<item type="gloss">be intermittently bursting out in shouts.</item>
<item type="gram">vi.i.pl</item>
<morphemes>
    <morph>
        <item type="text">sá'a</item>
        <item type="gloss">make:noise(PL)</item>
    </morph>
    <morph>
        <item type="text">k</item>
        <item type="gram">SGL</item>
    </morph>
    <morph>
        <item type="text">man</item>
        <item type="gram">POSTG</item>
    </morph>
    <morph>
        <item type="text">ta</item>
        <item type="gram">REP</item>
    </morph>
</morphemes></word>
```

3.7. Hausa

- SIL language code: HUA
- Source of data: Abdoulaye, Mahamane L. (1992) *Aspects of Hausa Morphosyntax in Role and Reference Grammar*, Ph.D. dissertation, SUNY at Buffalo. [http://linguistics.buffalo.edu/people/students/dissertations/abdoulaye/hausadiss.pdf]
- Size: ~595 example sentences (some of the examples are in languages other than Hausa)
- Original encoding: PDF
- Example of original encoding:

```
yaa sàa Indoo suuyàr gujiyaa.
3ms.PERF put Indo fry-DN-of peanuts
'He made Indo fry the peanuts.'
```

- Migration path: IGT extracted from original PDF document and converted to HBB (using methods described in Lewis 2003). XML errors corrected by hand and language codes added.
- Example of HBB encoding:

```
<phrase><item type="gloss">'He made Indo fry the peanuts.'</item>
<words>
    <word><item type="text">yaa</item>
    <morphemes>
        <morph>
            <item type="text">yaa</item>
            <item type="gram">3.m.s.PERF</item></morph></morphemes></word>
    <word><item type="text">sàa</item>
    <morphemes>
        <morph>
            <item type="text">sàa</item>
            <item type="gloss">put</item></morph></morphemes></word>
    <word><item type="text">Indoo</item>
    <morphemes>
        <morph>
            <item type="text">Indoo</item>
            <item type="gloss">Indo</item></morph></morphemes></word>
    <word><item type="text">suuyàr</item>
    <morphemes>
        <morph>
            <item type="text">suuyàr</item>
            <item type="gloss">fry</item></morph></morphemes></word>
    <word><item type="text">gujiyaa.</item>
    <morphemes>
        <morph>
            <item type="text">gujiyaa.</item>
            <item type="gloss">peanuts</item></morph></morphemes></word>
</words></phrase>
```

3.8. Passamaquoddy

- SIL language code: MAC
- Source of data: Bruening, Benjamin (2001) *Syntax at the Edge: Cross-Clausal Phenomena and the Syntax of Passamaquoddy*, PhD dissertation, MIT. [http://www.ling.udel.edu/bruening/Courses/2001-2/831/BrueningF2.pdf]
- Size: ~360 example sentences
- Original encoding: PDF
- Example of original encoding:

```
Kisi yaq ona skitapew-ehl-os-ultu-wok tan te etuci-woli-tahatomu-htit.
Able Quot also man-change.form-Refl-Plural-3P WH Emph IC.X.time-good-
   think.TI-3PConj
'They could, it is said, change themselves into men whenever it pleased
   them.' (Mitchell 1921/1976b, 16)
```

- Migration path: IGT extracted from original PDF document and converted to HBB (using methods described in Lewis 2003). XML errors corrected by hand and language codes added.
- Example of HBB encoding:

```
<phrase><item type="gloss">'They could, it is said, change themselves into
men whenever it pleased them.' (Mitchell 1921/1976b,16)</item>
<words>
    <word><item type="text">Kisi</item>
    <morphemes>
        <morph>
            <item type="text">Kisi</item>
            <item type="gloss">Able</item>
        </morph></morphemes></word>
    <word><item type="text">yaq</item>
    <morphemes>
        <morph>
            <item type="text">yaq</item>
            <item type="gram">Quot</item>
        </morph></morphemes></word>
    <word><item type="text">ona</item>
    <morphemes>
        <morph>
            <item type="text">ona</item>
            <item type="gloss">also</item>
        </morph></morphemes></word>
    <word><item type="text">skitapew-ehl-os-ultu-wok</item>
    <morphemes>
        <morph>
            <item type="text">skitapew</item>
            <item type="gloss">man</item></morph>
        <morph>
            <item type="text">ehl</item>
            <item type="gloss">change</item>
            <item type="gloss">form</item></morph>
        <morph>
            <item type="text">os</item>
            <item type="gloss">Refl</item></morph>
        <morph>
            <item type="text">ultu</item>
            <item type="gram">Plural</item></morph>
        <morph>
            <item type="text">wok</item>
            <item type="gram">3P</item></morph></morphemes></word>
    <word><item type="text">tan</item>
    <morphemes>
        <morph>
            <item type="text">tan</item>
            <item type="gram">WH</item></morph></morphemes></word>
    <word><item type="text">te</item>
    <morphemes>
        <morph>
            <item type="text">te</item>
```

```
            <item type="gram">Emph</item></morph></morphemes></word>
        <word><item type="text">etuci-woli-tahatomu-htit.</item>
        <morphemes>
            <morph>
                <item type="text">etuci</item>
                <item type="gram">IC</item>
                <item type="gram">X</item>
                <item type="gloss">time</item></morph>
            <morph>
                <item type="text">woli</item>
                <item type="gloss">good</item></morph>
            <morph>
                <item type="text">tahatomu</item>
                <item type="gloss">think</item>
                <item type="gram">TI</item></morph>
            <morph>
                <item type="text">htit.</item>
                <item type="gram">3PConj</item></morph></morphemes></word>
    </words></phrase>
```

4. Migration to the RDF database model using the metaschema

The HBB approach to the XML encoding of IGT provides a markup schema (that is, the formal definition of the permitted vocabulary and syntax of the XML markup). The GOLD ontology provides a semantic schema (that is, a formal definition of concepts in the problem domain). The migration of marked up language resources to an interoperable semantic representation in an RDF database is achieved by means of what Simons (2002, 2003) has termed a *metaschema*—a formal definition of how the elements and attributes of a markup schema are to be interpreted in terms of the concepts of a semantic schema. The Semantic Interpretation Language (SIL) was developed as a means of formally expressing how markup is to be interpreted in terms of concepts (Simons 2004).

The metaschema language and its operation are most easily explained by giving an example. For instance, consider the HBB-style markup shown in Figure 5 of a Hopi word and its interlinear analysis.

```
<word>
   <item type="text">aa'asnatoyni'ywisa</item>
   <item type="gloss">perform washing of the hair</item>
   <item type="gram">vt.i.pl</item>
</word>
```

Figure 5. Sample Hopi word with interlinear analysis

Figure 6 shows an extract from the metaschema that was used to interpret HBB-style markup in this project. An SIL metaschema, as described in detail in Simons (2004), is an XML document built from metaschema *directives;* each directive is essentially a processing instruction expressed as an XML element. The directives `resource`, `property`, and `literal` generate RDF resources, properties, and literals, respectively. Each of these uses a `concept` attribute to name the ontological concept of which the generated element is to be an instance. The `interpret` directive matches specific markup elements of the input resource and indicates how they are to be interpreted semantically. Thus, the first directive in Figure 6 declares that all occurrences of the `<word>` element are to be interpreted as instance of the GOLD concept *CompleteLexicalUnit*. LinguisticUnits in GOLD (of which CompleteLexicalUnit is a subclass) are analyzed as having three primary properties, *form*, *meaning*, and *grammar*. The remaining `interpret` directives in

Figure 6 instruct that `<item>` elements with `type` attributes of “text”, “gloss”, and “gram” are to be interpreted as instances of these three properties, respectively. Furthermore, these `<item>` elements also create resources of types *LinguisticForm*, *CompositionalSense*, and *MorphosyntacticInformation*, respectively. These in turn may have a number of properties (though in each of these cases, only one of the possible properties is used).

```
<interpret markup="word">
  <resource concept="gold:CompleteLexicalUnit"/>
</interpret>
<interpret markup="item[@type='text']">
  <property concept="gold:form">
    <resource concept="gold:LinguisticForm">
      <literal concept="gold:symbolicRepresentation"/>
    </resource>
  </property>
</interpret>
<interpret markup="word/item[@type='gloss']">
  <property concept="gold:meaning">
    <resource concept="gold:CompositionalSense">
      <literal concept="gold:translation"/>
    </resource>
  </property>
</interpret>
<interpret markup="item[@type='gram']">
  <property concept="gold:grammar">
    <resource concept="gold:MorphosyntacticInformation">
      <property concept="gold:features">
        <collection>
          <resourceRef tokenize="." namespace="#"/>
        </collection>
      </property>
    </resource>
  </property>
</interpret>
```

Figure 6. Metaschema directives for interpreting markup in Figure 5

Applying the metaschema in Figure 6 to the language resource in Figure 5 results in the RDF/XML document shown in Figure 7. When the metaschema processor creates a new instance of an RDF resource, it automatically creates a unique identifier for it. Thus “#element(121/1)” is an identifier that is unique within the document; it happens to indicate the first child element of the 121st element of the input document. Of particular interest is the interpretation of the gram string, “vt.i.pl”. The metaschema directive instructs that it should be tokenized by splitting on the “.” character, and that the result should be a collection of references to existing RDF resources, where each extracted substring (i.e. *vt*, *i*, and *pl*) is the unique identifier of an existing resource. The notation `<rdf:Description rdf:about="#vt"/>` is the RDF/XML syntax for expressing a reference to the resource named *vt* in the current document. In this case, that resource is the Hopi-specific instance of the GOLD concept *TransitiveVerb* that is defined by the Hopi language profile. The language profile is also interpreted by means of a metaschema in order to create this and the other RDF resources that the grammatical analyses of words and morphemes refer to.

```
<rdf:RDF xml:base="http://emeld.org/Hopi_examples">
...
<gold:CompleteLexicalUnit rdf:about="#element(121)">
 <gold:form>
```

```
      <gold:LinguisticForm rdf:about="#element(121/1)">
         <gold:symbolicRepresentation>aa'asnatoyni'ywisa
         </gold:symbolicRepresentation>
      </gold:LinguisticForm>
   </gold:form>
   <gold:meaning>
      <gold:CompositionalSense rdf:about="#element(121/2)">
         <gold:translation>perform washing of the hair</gold:translation>
      </gold:CompositionalSense>
   </gold:meaning>
   <gold:grammar>
      <gold:MorphosyntacticInformation rdf:about="#element(121/3)">
         <gold:features rdf:parseType="Collection">
          <rdf:Description rdf:about="#vt"/>
          <rdf:Description rdf:about="#i"/>
          <rdf:Description rdf:about="#pl"/>
         </gold:features>
      </gold:MorphosyntacticInformation>
   </gold:grammar>
</gold:CompleteLexicalUnit>
...
</rdf:RDF>
```

Figure 7. RDF/XML output for sample Hopi word

When an RDF/XML document is loaded into an RDF database such as Sesame, the database engine converts it into the equivalent set of Object-Attribute-Value triples. Figure 8 shows the result of translating the RDF/XML interpretation of the analyzed Hopi word into RDF triples. Each line of the figure contains one triple, expressed as the unique identifier of the subject resource, the predicate, and the object resource, separated by spaces. The prefix `hopi:` is an abbreviation for “http://emeld.org/Hopi_examples#”.At the end of Figure 8, note that the Collection from Figure 7 is automatically converted to a List structure by the RDF database; the resource identifiers beginning with “_:b” are automatically generated “blank node” identifiers.

```
Object              Attribute          Value

hopi:element(121)   rdf:type           gold:CompleteLexicalUnit
hopi:element(121)   gold:form          hopi:element(121/1)
hopi:element(121/1) rdf:type           gold:LinguisticForm
hopi:element(121/1) gold:symbolic-     "aa'asnatoyni'ywisa"
                    Representation
hopi:element(121)   gold:meaning       hopi:element(121/2)
hopi:element(121/2) rdf:type           gold:CompositionalSense
hopi:element(121/2) gold:translation   "perform-washing-of-the-hair"
hopi:element(121)   gold:grammar       hopi:element(121/3)
hopi:element(121/3) rdf:type           gold:MorphosyntacticInformation
hopi:element(121/3) gold:features      _:b76495
_:b76495            rdf:first          hopi:vt
_:b76495            rdf:rest           _:b76496
_:b76496            rdf:first          hopi:i
_:b76496            rdf:rest           _:b76497
_:b76497            rdf:first          hopi:pl
_:b76497            rdf:rest           rdf:nil
```

Figure 8. The semantic interpretation as RDF triples

5. Leveraging the database structure to enhance search

In this section, we present some queries and results over our RDF database using SeRQL (Broekstra, Kampman and van Harmelen, 2002), which includes the results of applying the metaschema to all the language profiles, and the following IGT data: Archi, Korean (portion), Mutsun (portion), Monguor, Hausa and Passamaquoddy. We loaded only about 12% of the Korean and 33% of theMutsun IGT data, and did not load any of the Hopi IGT data because of the difficulty Sesame had with their size. We also decided not to use the Hausa IGT data because of some remaining errors in the XML encoding.

The query in Figure 9 asks which languages use the features *Number* and *Case*. The results are shown in Figure 10; the language profiles for which both features are not present are Korean and Passamaquoddy.

```
select distinct Language
from {DV} <gold:varietyOf> {Language};
          <gold:defines> {} <rdf:type> {<gold:NumberAttribute>},
     {DV} <gold:defines> {} <rdf:type> {<gold:CaseAttribute>}
```

Figure 9. SeRQL query for languages with both Number and Case features

```
Language
http://ethnologue.com/SIL-language-code#ARC
http://ethnologue.com/SIL-language-code#MJG
http://ethnologue.com/SIL-language-code#HUA
http://ethnologue.com/SIL-language-code#CSS
http://ethnologue.com/SIL-language-code#HOP
5 results found in 262 ms.
```

Figure 10. Results of query in Figure 9

Next, the query in Figure 11 asks which words contain parts, one of which is marked *SecondPerson* and the other (possibly the same) *ImmediatePastTense*. Only one example was found in the data set, and it is a Passamaquoddy word. Figure 12 shows the literal search result, followed by the IGT in the original PDF file that corresponds to it. This result was found because the Passamaquoddy language profile indicates that '2' is to be interpreted as *SecondPerson* and 'Pret' as *ImmediatePastTense*.

```
select distinct Word
from {Word} <gold:grammar> {MSI};
          <rdf:type> {<gold:CompleteLexicalUnit>},
     {M1} <gold:constituentOf> {MSI};
          <gold:grammar> {M1_MSI},
     {F1} <gold:featureOf> {M1_MSI};
          <rdf:type> {<gold:ImmediatePastTense>},
     {M2} <gold:constituentOf> {MSI};
          <gold:grammar> {M2_MSI},
     {F2} <gold:featureOf> {M2_MSI};
          <rdf:type> {<gold:SecondPerson>}
```

Figure 11. Complex query involving *SecondPerson* and *ImmediatePastTense* features.

```
Word
http://emeld.org/Passamaquoddy_examples#element(235/2/1/1/2/4)
1 results found in 43 ms.
```

```
Keq=apc sesolahki=te mihqitahas-iyin ehcuwi-monuhmon-s?
what=again suddenly=Emph remember-2Conj IC.must-buy.2Conj-DubPret
'What else did you suddenly remember you had to buy?' AH,SN 8:5.8
```

Figure 12. Result of query in Figure 11

Finally, the query in Figure 13 asks which *language-specific* feature values appear in word-final morphs. The results are given in Figure 14 (with namespaces omitted). Note that the feature and value names are those of the language profile, not their corresponding GOLD concepts.

```
select distinct Feature, Value, Language
from {DV} <gold:varietyOf> {L};
          <gold:defines> {F} <gold:possibleValue> {V},
     {LI} <rdf:rest> {<rdf:nil>};
          <rdf:first> {V} <gold:featureOf> {MSI} <gold:__features> {LI},
     {M} <gold:grammar> {MSI};
         <rdf:type> {<gold:SublexicalUnit>}
```

Figure 13. Query for word-final features

Feature	*Value*	*Lang*
case	COMPAR	ARC
case	CONTABL	ARC
case	CONTALL	ARC
case	CONTLAT	ARC
case	EQU	ARC
case	ERG	ARC
case	GEN	ARC
case	IMPERF	ARC
case	INABL	ARC
case	INALL	ARC
case	INESS	ARC
case	INTERABL	ARC
case	INTERALL	ARC
case	INTERESS	ARC
case	INTERLAT	ARC
case	INTERTRANS	ARC
case	INTRANS	ARC
case	NEUTER	ARC
case	NOM	ARC
case	PERM	ARC
case	SUBABL	ARC
case	SUBALL	ARC
case	SUBESS	ARC
case	SUBLAT	ARC
case	SUBTRANS	ARC
case	SUPERABL	ARC
case	SUPERALL	ARC
case	SUPERESS	ARC
case	SUPERLAT	ARC
case	SUPERTRANS	ARC
number	SG	ARC
number	PL	ARC
unassigned	CLASSI	ARC
unassigned	FIN	ARC
unassigned	OBL	ARC
Case	ACC	MJG
Case	CAUS	MJG
Case	DAT	MJG

Feature	*Value*	*Lang*
Case	DIR	MJG
Case	GEN	MJG
Aspect	PERF	MJG
Aspect	IMPF	MJG
Aspect	PROG	MJG
Noun-Attribute	N1	MJG
Noun-Attribute	N2	MJG
Noun-Attribute	Npr	MJG
Noun-Attribute	Ntemp	MJG
Quantifier	Nu	MJG
Number	sg	MJG
Modality	PURP	MJG
Modality	QUOT	MJG
Voice	COND	MJG
Tense	FUT	MJG
Transitivity	vt	MJG
Transitivity	Vt	MJG
Transitivity	Vi	MJG
Pronoun-Attribute	PNind	MJG
Particle-Attribute	POSS	MJG
unassigned	#	MJG
unassigned	NZR	MJG
mode	Pret	MAC
mode	Sub	MAC
voice	Dir	MAC
voice	Inv	MAC
voice	Recip	MAC
gender	An	MAC
gender	Inan	MAC
person	1	MAC
person	2	MAC
person	3	MAC
person	Obv	MAC

Feature	*Value*	*Lang*
person	12	MAC
person	Indef	MAC
number	P	MAC
number	Plural	MAC
aspect	Fut	MAC
aspect	Perf	MAC
aspect	Prog	MAC
order	Conj	MAC
order	Imp	MAC
n_unassigned	Loc	MAC
v_unassigned	App	MAC
v_unassigned	IC	MAC
v_unassigned	N	MAC
v_unassigned	Neg	MAC
case	ABLATIVE	CSS
case	ANDATIVE	CSS
case	BENEFACTIVE	CSS
case	CAUSATIVE	CSS
case	INSTRUMENTAL	CSS
case	LOCATIVE	CSS

Feature	*Value*	*Lang*
case	PERSONAL_LOCATIVE	CSS
case	VENITIVE	CSS
number	PLURAL	CSS
Aspect	CONTINUATIVE	CSS
Aspect	INTENSIVE	CSS
Aspect	PERFECTIVE	CSS
Modality	IMPERATIVE	CSS
Voice	REFLEXIVE	CSS
Voice	RECIPROCAL	CSS
Voice	MEDIOPASSIVE	CSS
Voice	PASSIVE	CSS
Tense	RECENT_PAST_TENSE	CSS
Tense	REMOTE_PAST_TENSE	CSS
unassigned	OBJECTIVE	CSS
unassigned	NOMINALIZER	CSS
unassigned	POSITIONAL_-CAUSATIVE	CSS
case	PCA	KKN
mood	EFN	KKN
tense	EPF	KKN

135 results found in 240 ms.

Figure 14. Results of query in Figure 13

References

Bell, John and Steven Bird (2000). A preliminary study of the structure of lexicon entries. *Workshop on Web-Based Language Documentation and Description*, Philadelphia. [www.ldc.upenn.edu/exploration/expl2000/papers/bell/bell.html]

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. *Scientific American*, May 2001.

Bird, Steven and Gary F. Simons, (2003). Seven dimensions of portability for language documentation and description. *Language* 79(3):557-582.

Brickley, Dan and R. V. Guha, eds. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. Technical report, W3C. [http://www.w3.org/TR/rdf-schema/]

Broekstra, J., A. Kampman, and F. van Harmelen (2002). Sesame: A generic architecture for storing and querying RDF and RDF schema. *Proceedings of the First International Semantic Web Conference* , I. Horrocks and J. Hendler, ed., pages 54-68, Springer-Verlag, Berlin.

Chamberlin, Don, Daniela Florescu, Jonathan Robie, Jerome Simeon and Mugur Stefanescu (2001). *XQuery: A Query Language for XML*. Working draft, World Wide Web Consortium. [http://www.w3.org/TR/xquery/].

Farrar, Scott and D. Terence Langendoen (2003) A linguistic ontology for the Semantic Web. *GLOT International* 7(3):97-100. [http://emeld.org/documents/GLOT-LinguisticOntology.pdf]

Farrar, Scott, William D. Lewis and D. Terence Langendoen (2002). An ontology for linguistic annotation. *Semantic Web Meets Language Resources: Papers from the AAAI Workshop, Technical Report WS-02-16*, pages 11-19. Menlo Park, CA: AAAI Press. [http://emeld.org/documents/AAAI-OntologyLinguisticAnnotation.pdf]

Hughes, Baden, Steven Bird and Cathy Bow (2003). Interlinear text facilities. Demonstrated at *Workshop on Digitizing and Annotating Texts and Field Recordings,* LSA Institute,

Michigan State University. [http://emeld.org/workshop/2003/baden-demo.html]. See also [http://www.cs.mu.oz.au/research/lt/emeld/interlinear].

Karvounarakis, Gregory, Vassilis Christophides, Dimitris Plexousakis, and Soa Alexaki (2000). *Querying community web portals*. Technical report, Institute of Computer Science, FORTH, Heraklion, Greece. [http://www.ics.forth.gr/proj/isst/RDF/RQL/rql.pdf].

Lewis, William D. (2003). Mining and migrating interlinear glossed text. *Workshop on Digitizing and Annotating Texts and Field Recordings*, LSA Institute, Michigan State University. [http://emeld.org/workshop/2003/paper-lewis.html]

McGuinness, D. L. and F. van Harmelen, eds. (2004). OWL Web Ontology Language overview. Technical report, W3C. [http://www.w3.org/TR/2004/REC-owl-features-20040210]

Seaborne, Andy (2004). *RDQL - A query language for RDF*. Technical report, HP Labs Bristol. W3C Member Submission. [http://www.w3.org/Submission/RDQL/]

Simons, Gary F. (1997). PTEXT: A format for the interchange of parsed texts among natural language processing applications. *SIL Electronic Working Papers 1997-008*. [http://www.sil.org/silewp/1997/008/SILEWP1997-008.html]

Simons, Gary F. (2002). The electronic encoding of lexical resources: A roadmap to Best Practice. *EMELD Workshop on Digitizing Lexical Information*, Ypsilanti, MI. [http://emeld.org/documents/roadmap.htm]

Simons, Gary F. (2003). Developing a metaschema language to support interoperation among XML resources with different markup schemas. Paper presented at the ACH/ALLC conference, Athens, GA.

Simons, Gary F. (2004). A metaschema language for the semantic interpretation of XML markup in documents. Technical report, SIL, Dallas. [http://www.sil.org/~simonsg/metaschema/sil.htm]

Simons, Gary F., William D. Lewis, Scott Farrar, D. Terence Langendoen, Brian Fitzsimons and Hector Gonzalez (2004). The semantics of markup: Mapping legacy markup schemas to a common semantics. *Proceedings of NLPXML 2004, RDF/RDFS and OWL in Language Technology: 4th Workshop on NLP and XML*, Barcelona, Spain, 2004-07-26. [http://emeld.org/documents/SOMFinal1col.pdf]

User Guide for Sesame (2004). [www.openrdf.org/publications/users/index.html]