OLIF v.2: A Flexible Language Data Standard

Susan M. McCormick
smccormick@comcast.net

Christian Lieske
christian.lieske@sap.com

Alexander Culum
a_culum@yahoo.com

Abstract

The Open Lexicon Interchange Format (OLIF) v.2 is an open standard for users of language technology. XML-compliant and freely available to the community, it is an exchange format with both lexical and terminology application that can address language data management needs in an environment increasingly concerned with global product development. Designed to facilitate general language data exchange, OLIF is also specifically equipped to cover more detailed linguistic requirements for six European languages. In addition to a representative array of administrative, morphological, syntactic, and semantic data categories, OLIF offers a modeling of transfer restrictions for representing context-dependent transfer statements. The flexibility and user extensibility of the XML Data Type Definition (DTD) implementation of OLIF are expanded on and improved in the soon-to-be-released OLIF XML Schema Definition Language (XSD) implementation.

I. Introduction

The Open Lexicon Interchange Format (OLIF) is an XML-compliant lexical/terminological exchange format specifically designed to streamline the exchange of language data for users of multiple language technology tools. Implemented and supported by the OLIF2 Consortium, version 2 of OLIF is open and available to the public from the consortium web site **www.olif.net**. Visitors to the site may download free-of-charge the OLIF v.2 specification, the official OLIF v.2 XML Data Type Definition (DTD), descriptions of all data categories, elements and attributes, sample XSL stylesheets, and suggested guidelines for standard formulations of entry words and phrases.

Although the original intent of OLIF was to provide for the exchange of lexical data between proprietary machine translation (MT) lexicons, users will note that it has evolved into a more general data exchange standard for the educated language technology user.

1.1 Early development of OLIF

The prototype for the OLIF format was generated in the mid 1990's as an integral part of the OTELO project, an EC-funded group of industrial language technology developers and users. The primary goal of OTELO was to develop interfaces and formats that would help users meet the challenges of translation and localization by better leveraging existing language tools. Many of the OTELO partners were increasingly facing the problem of managing large stores of lexical and terminological data in diverse applications (e.g., MT, translation memory (TM) and terminology databases) with idiosyncratic and often proprietary formats.

While it was generally recognized that the language tools offered a great potential for productivity gains, the demands for manual labor in the area of lexical and terminology data management were often overwhelming many of the gains that were made. Getting company-specific terminology into an MT lexicon, for instance, was usually a time-consuming process that required expert help from either the vendor or trained in-house linguists. Available terminology exchange standards such as the Machine-Readable Terminology Interchange Format (MARTIF; http://www.iso.org/iso/en/) were generally not supported because their structure and content were not geared to the lexical model that was reflected in the MT lexicons.

What was needed, it was thought, was a standard format that would support enough linguistic representation to accommodate language applications like MT, but also respond to terminology requirements. A format that facilitated data exchange among different MT systems, but also provided for the transfer of terminology from a terminology database to an MT lexicon would go a long way toward alleviating the data exchange bottleneck that was hampering the use of otherwise helpful language tools.

With this in mind, the OTELO project developed an OLIF prototype based on the requirements of project member MT lexicons and terminology databases. The prototype represented a relatively flat format in structure and included generous coverage of English and German lexical data categories. The basic structure was a monolingual set of lexical/terminological data categories with links for transfer and cross-reference. As a first attempt, the original OLIF succeeded in providing an alternative to the traditional time- and people-intensive method of updating MT lexicons with terminology. It was also apparent, however, that the OLIF prototype was really a jumping-off point; what was really needed was a standard that supported other language technologies, not just MT, and was much more inclusive in terms of the actual languages covered. In addition, making OLIF XML-compliant would greatly improve its value as a language data exchange option.

1.2 The OLIF2 Consortium

Recognizing the potential benefits of a revised OLIF, SAP founded the OLIF2 Consortium in March 2000 with the singular purpose of developing the *Open Lexicon Interchange Format version 2* (OLIF v.2, or OLIF2). Joining the consortium were language technology companies and organizations such as Xerox, Microsoft, Trados, IBM, Systran, IAI, DFKI and Comprendium. Consortium members were tasked with using the original OLIF prototype to design an XML-compliant language data exchange format that would enable the exchange of basic lexical/terminological data for a wide range of strategic languages. To accommodate users of complex language technologies such as MT, OLIF v.2 would also provide expanded linguistic support for six major European languages (English, German, Spanish, French, Portuguese, and Danish). In addition, the OLIF2 Consortium would contribute OLIF v.2 knowledge for the lexicographical component of the XLT lexical/terminology exchange standard that was being designed by the SALT project (http://www.ttt.org/salt/index.html).

The result of the work of the OLIF2 Consortium was released in February 2002 as OLIF v.2 and is currently available to the community from the consortium Web site. In structure and content, OLIF v.2 represents a useful niche between MARTIF-type formats such as TBX (http://www.lisa.org/tbx/), and proprietary formats for formalized Natural Language Processing (NLP) development needs. It is intended for *users* of language technology, especially users who may require a more expansive linguistic analysis than standard terminology exchange formats offer. The expanded linguistic analysis is reflected in both the lexical view of data that OLIF v.2 supports and the data categories that it defines. Given the flexibility and breadth of coverage described in the following pages, however, readers will also note that, in addition to providing for exchange of more detailed linguistic information, OLIF v.2 is a valid, general option for a wide array of language data management jobs, including simple terminology exchange.

2. The structure of an OLIF entry

Since the original goal of OLIF was to provide a bridge between MT lexicons and terminology management applications, it must take into account both a lexical and terminological view of the data. The resulting structure of OLIF entries is accordingly a hybrid model that is neither explicitly lemma-oriented, as many dedicated lexicons are, nor explicitly concept-oriented, as many terminology management models are, i.e., with formal concept and term levels. It is perhaps most helpful to describe the OLIF model as word-sense-oriented. For the purposes of describing OLIF, this can be taken to mean that an entry is defined as a collection of monolingual data on a specified sense of the word or phrase, with optional links to represent transfer and cross-reference relations.

2.1 The OLIF mono

The monolingual data (mono) within an entry is grouped according to the linguistic/lexical/terminological character of the information being represented. The groups themselves are sub-lists of data category/value pairs[1]. For example, a typical OLIF v.2 entry might encode information on the English noun *table* with data groupings like *key, administrative, morphological, syntactic,* and *semantic* (see Figure 1).

```
<entry>
   <mono>
      <keyDC>
        <canForm>table</canForm>
        <language>en</language>
        <ptOfSpeech>noun</ptOfSpeech>
        <subjField>general</subjField>
        <semReading>86</semReading>
      </keyDC>
      <monoDC>
         <monoAdmin>
           <originator>Weber</originator>
           <adminStatus>ver</adminStatus>
         </monoAdmin>
         <monoMorph>
           <inflection>like book,books</inflection>
         </monoMorph>
         <monoSyn>
           <synType>cnt</synType>
           <synFrame>[gencomp-opt]</synFrame>
         </monoSyn>
         <monoSem>
            <definition>An arrangement of words, numbers, or signs or
            combinations of them, as in parallel columns, to exhibit a set of
            facts or relations in a definite, compact, and comprehensive
            form.</definition>
            <semType>inform</semType>
         </monoSem>
      </monoDC>
    </mono>
</entry>
```

Figure 1: The OLIF mono

The key data categories identify the mono uniquely and include data on *canonical form, language, part of speech, subject field,* and *semantic reading*; the administrative data categories contain information used to organize or identify the mono administratively (e.g., *originator, administrative status, geographical usage*); the morphological data categories describe the morphological structure and status of the monolingual string (e.g., *inflection gender*); the syntactic data categories refer to the syntactic behavior associated with the mono (e.g., *syntactic type, syntactic frame*); and the semantic data categories represent information on the semantic analysis for the mono (e.g., *semantic type, natural gender*).

[1] The data category/value pairs are represented in XML as tags that reflect the element types, attributes, and values defined in the XML DTD/schema

2.2 Transfer and cross-reference in OLIF

While the mono element contains data that refers to the status and behavior of the entry string, the *transfer* and *cross reference* elements represent links for the given mono to other entries; a transfer points to an entry in another language, whereas a cross-reference points to an entry in the same language.

Transfer in OLIF is essentially defined as bilingual and unidirectional, that is, each transfer group in an entry 1) refers to a single link between two entries in different languages, and 2) implies a transfer from the source (i.e., the entry described in the mono) to the target (i.e., the entry described in the transfer). An OLIF entry may contain an unlimited number of transfer elements, meaning that the lexicographer can specify multiple transfers to the same language (e.g., English *source* -> German *target1*, German *target2*…), and/or multiple transfers into different languages (e.g., English *source* -> German *target*, French *target*, Spanish *target*…). Restrictions on the scope of a transfer (e.g., *source x is target y in context z*) are represented in the transfer element of OLIF by means of *transfer restrictions* (see section 3.1).

The semantics of cross-reference in OLIF also imply a link with directionality from the mono to the entry that is being referred to. For instance, the entry for English *table* may contain a cross-reference to the entry for English *row* via the cross-reference relation *has-meronym*, meaning that *table* is a whole which has a part *row*. Correspondingly, the entry for English *row* may contain a cross-reference to *table* for the relation *has-holonym*, indicating that it is a part of the whole *table*. OLIF transfer and cross-reference are illustrated in Figure 2:

```
<entry>
  <mono>
     <keyDC>
       <canForm>table</canForm>
       <language>en</language>
       <ptOfSpeech>noun</ptOfSpeech>
       <subjField>general</subjField>
       <semReading>86</semReading>
     </keyDC>
     <monoDC>
        <monoAdmin>
          <originator>Weber</originator>
          <adminStatus>ver</adminStatus>
        </monoAdmin>
        <monoMorph>
          <inflection>like book,books</inflection>
        </monoMorph>
        <monoSyn>
          <synType>cnt</synType>
          <synFrame>[gencomp-opt]</synFrame>
       </monoSyn>
       <monoSem>
          <definition>An arrangement of words, numbers, or signs or
         combinations of them, as in parallel columns, to exhibit a set of
         facts or relations in a definite, compact, and comprehensive form.
          </definition>
          <semType>inform</semType>
       </monoSem>
     </monoDC>
   </mono>

     <crossRefer>
       <keyDC>
          <canForm>row</canForm>
          <language>en</language>
          <ptOfSpeech>noun</ptOfSpeech>
          <subjField>general</subjField>
          <semReading>69</semReading>
       </keyDC>
       <crLinkType>has-meronym</crLinkType>
     </crossRefer>
     <transfer>
       <keyDC>
          <canForm>Tabelle</canForm>
          <language>de</language>
          <ptOfSpeech>noun</ptOfSpeech>
          <subjField>general</subjField>
          <semReading>86</semReading>
       </keyDC>
     </transfer>

</entry>
```

Figure 2: OLIF entry with cross-reference and transfer

Since the specification of cross-reference and transfer links is optional, a minimal well-formed OLIF v.2 entry contains a mono group with the key data categories *canonical form, language, part of speech, subject field,* and *semantic reading,* which, as noted above, together serve to identify the entry uniquely. Users may find minimally-specified OLIF entries a useful alternative to simple comma-separated formats or similar skeletal modelings of term entries. The relatively flat format of OLIF means that basic entries are fairly easy to generate and read. In addition, the optional morphological, syntactic and semantic OLIF data categories provide the user with choices for a more robust lexical/terminological description.

The reader will note in section 4.1 that the OLIF specification also provides for an even more efficient representation by offering the option of a numeric identifier for the mono or key data categories. Either of these IDs can be used in place of the list of five key data categories in any transfer or cross-reference component to identify the mono that is being linked to.

3. The content of an OLIF entry

The original aim of OLIF was to provide a description of a lexical/ terminological entry to the extent that an NLP vendor could generate a basic, usable entry of its own from an OLIF record. With this in mind, developers reviewed existing terminology exchange formats, as well as the formats of existing MT systems for European languages, trying to identify commonality in the content of what was being represented. For instance, can a common representation for a data category like *semantic type* be offered in OLIF, or is the analysis of this category so theory- or system-specific that an attempt at a unitary description brings very little in the way of a return?

Since the goal was to make it possible for language technologists to map to OLIF, the OLIF data categories needed to be as generic as possible, both in the choice of data categories themselves and the values that are defined for the data categories. For example, OLIF supports designators from ISO 639 1 as values for the data category *language,* as well as the use of the standard *xml:lang.* While paying attention to general applicability, though, there also had to be enough specific linguistic coverage to provide an exchange of the data required for an adequate lexical entry.

The result for an OLIF v.2 monolingual entry is a format that supports, in addition to the basic key defining data categories, 19 administrative data categories, 12 morphological data categories, 7 syntactic data categories, and 3 semantic data categories. In all cases, the data category and associated values are intended to have general application for the targeted languages.

3.1 A typology of data categories and values in the OLIF entry

The administrative data categories in OLIF were derived from a review of language technology users in the OLIF Consortium as well as a comparison with the metadata analysis of the terminology standard ISO 12620 (http:www.ttt.org/clsframe/datcats.html). The administrative data categories are separated into two groups in OLIF:

1. data categories that apply only to the *mono*
2. data categories that apply to all three major sub-groups of the entry, i.e., *mono, cross-reference, transfer*

The data categories in group 1 include options for *syllabification, geographical usage (dialect), entry type, phrase type, entry source, originator (author),* and *company.* The more generally applicable data categories in group 2 include *updater (editor), modification date, usage, note,* and *example*. In all cases, the values associated with the data categories were brought as closely in line with the values specified in ISO 12620 as possible.

Linguistic data categories in OLIF were culled from MT systems such as Comprendium and Logos and were vetted for general linguistic validity; data categories that were idiosyncratic to a particular system were usually viewed as issues for that particular system's converter. As mentioned already, the linguistic data categories also fell into distinguishable groups, which have been reflected in the XML implementation provided by the consortium:

1. morphological data categories for the *mono*
2. syntactic data categories for the *mono*
3. semantic data categories for the *mono*
4. transfer restriction data categories for the *transfer* element
5. data categories for cross-reference linking in the *cross-reference* component

In trying to define values for the linguistic data categories, the basic metric was that they be ‘mappable’ to different NLP lexicons. Although lexicon formats from systems like MT show great variation in structure and content, there is also a definable agreement in the essence of what is being represented. This agreement indicates both the general commonality of language structure in languages around the world, and the similarity of structure in languages that are related to one another. In settling on analyses and representations for the linguistic data categories and their values, the consortium attempted to define this area of agreement, thus maximizing the chances that different system formats would be mappable to and from OLIF.

While the data categories selected for OLIF are themselves all generally supported within the field, developers found a range of agreement on how different data categories should be represented in terms of the values that were to be assigned to it:

- For some data categories, there was widespread agreement on possible values; these include categories like *grammatical gender, case, number, person,* and *tense,* where the analysis for European languages has a long history and is widely accepted. Other standardization efforts such as EAGLES (http://www.ilc.cnr.it/EAGLES) were especially helpful in these cases for producing the final set of values for each data category.
- Well-represented data categories that are modeled in theory- or system-specific ways were more resistant to a unitary OLIF representation. In these cases, there was an attempt to identify standards that enjoy support in the community and adapt them. For example, for the data category *subject field (subjField)*, the EC, an OLIF partner, offered the basic subject fields from Eurodicautom[2] (http://europa.eu.int/eurodicautom):

VALUE	**DESCRIPTION**
agriculture	farming and agriculture
audiovisual	audiovisual
aviation	aviation and aerospace
botany/zoology	botany and zoology
budget	budgets and accounting
chemistry	chemistry
construction	construction and building
customs	customs, duties
defense	defense
development	development
economics	economics
education	education
electrotechnics	electronics
….	

Figure 3: Sample of subject field values from Eurodicautom

[2] The basic subject field values for Eurodicautom have been updated since the publication of version 2 of OLIF. The updated values are currently being incorporated into OLIF v.2 as part of a package of small upgrades.

Similarly, to define the set of word relation types for the data category *cross-reference link type (crLinkType)*, the analysis of the EuroWordNet project (www.illc.uva.nl/EuroWordNet) was adapted:

VALUE	**DESCRIPTION**
synonym	synonym of
near-synonym	near synonym of
antonym	antonym of
near-antonym	near antonym of
has-hyperonym	is kind of (subordinate)
has-hyponym	has kind (superordinate)
has-holonym	part of
has-meronym	whole of
has-holo-member	member of (member-set)
has-mero-member	set (member-set)
has-holo-portion	portion of
has-mero-portion	has portion
has-holo-madeof	ingredient of
has-mero-madeof	has ingredient
has-holo-location	more specific place
has-mero-location	wider place
causes	cause of
is-caused-by	effect of
has-subevent	(between verbs/gerunds) *e.g., sleep ~ snore*
is-subevent-of	(between verbs/gerunds) *e.g., snore ~ sleep*
role	activity that something (noun) is involved in
involved	thing (noun) involved in activity represented by verb
….	

Figure 4: Sample of OLIF values for *crLinkType* adapted from EuroWordNet

- With several data categories, there was agreement among participating systems on the basics of an analysis, but how the basics were combined or interpreted varied from system to system. In these cases, the OLIF designers attempted to identify the basics of the analysis, the building blocks, so to speak, and provide these as values; using the building blocks, an MT system converter, for instance, could create an OLIF analysis that is compatible with the MT system's.

  For example, the data category ***syntactic frame (synFrame)****,* which contains data on subcategorization for an entry has various corollaries in different MT systems, ranging from explicit frame representations to system-specific codes that fuse semantic distinctions with their syntactic reflexes. Rather than adopting completely one of these approaches, OLIF developers took the lead from Slot Grammar (see, e.g., McCord, 1980; McCord, 1982) and defined frame elements as building blocks for a syntactic frame description; a simple syntax for putting the elements together in a frame was then defined. Using the frame elements and combining syntax, a lexicographer could represent a syntactic frame for the English verb *try* in OLIF in a straightforward, hopefully mappable, way:

[subj, (dobj-opt | dobj-sent-ing-opt | dobj-sent-inf-opt)]

"English *try* subcategorizes a subject and optional direct object noun phrase, *ing*-clause, or infinitive clause."

Figure 5: Sample value for *synFrame*

The issue of ***transfer restrictions***, where the lexicographer specifies a context in which a transfer is valid, also lent itself to the 'building-block' approach. Here, OLIF developers broke down the representation of transfer restrictions into two isolable units of representation:

1) The *context(s)* in the source language for a given translation of a source word or phrase into a target word or phrase.
2) *Test(s)* on the data categories/values associated with the context

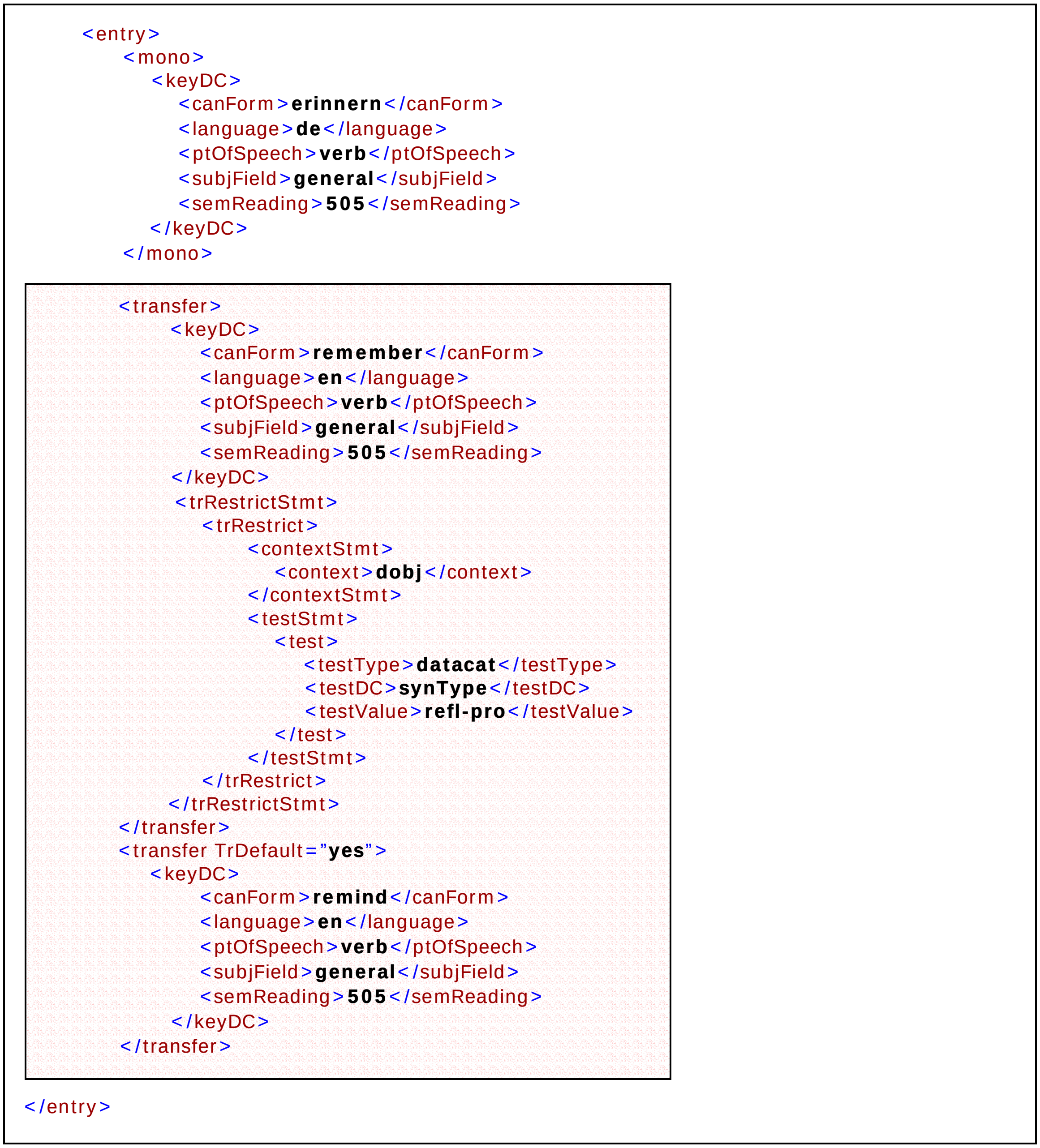
The context was analyzed further as consisting of several types:

a) The source word/phrase itself
b) Distinct context elements that occur with the source word/phrase within the clause. (These elements usu. fall within the syntactic frame defined for that particular word/phrase.) The context elements are generally categorized based on their part-of-speech.
c) Phrases that must be matched word-for-word for the condition to be satisfied, e.g., *trip the light fantastic, be in hot water.*

Tests on the different context types were identified as two simple types:

a) Tests on data category values.
b) Tests on specified strings.

In addition, a *logical operator* data category was defined to allow the combining of contexts and tests in maximally expressive ways. In breaking the representation of transfer restrictions down to these few basic elements, OLIF provides a means of mapping a system-specific rendering of transfer restrictions to one that could be construed by a system with a different analysis and format. Figure 6 shows an example of the transfer of the German verb *erinnern* to English:



```
    <entry>
       <mono>
          <keyDC>
             <canForm>erinnern</canForm>
             <language>de</language>
             <ptOfSpeech>verb</ptOfSpeech>
             <subjField>general</subjField>
             <semReading>505</semReading>
          </keyDC>
       </mono>

        <transfer>
            <keyDC>
               <canForm>remember</canForm>
               <language>en</language>
               <ptOfSpeech>verb</ptOfSpeech>
               <subjField>general</subjField>
               <semReading>505</semReading>
            </keyDC>
            <trRestrictStmt>
               <trRestrict>
                   <contextStmt>
                      <context>dobj</context>
                   </contextStmt>
                   <testStmt>
                      <test>
                         <testType>datacat</testType>
                         <testDC>synType</testDC>
                         <testValue>refl-pro</testValue>
                      </test>
                   </testStmt>
               </trRestrict>
            </trRestrictStmt>
        </transfer>
        <transfer TrDefault=”yes”>
           <keyDC>
               <canForm>remind</canForm>
               <language>en</language>
               <ptOfSpeech>verb</ptOfSpeech>
               <subjField>general</subjField>
               <semReading>505</semReading>
            </keyDC>
        </transfer>

</entry>
```

Figure 6: Transfer of German *erinnern* to English

The transfer component in Figure 6 says that German *erinnern* is to be translated as English *remember* in the context of a source direct object that is a reflexive pronoun; otherwise, *erinnern* is to be assigned an English transfer of *remind*. With the logical operator element (*logOp)* (not pictured), the user can state multiple transfer restrictions within the transfer restriction statement, thus significantly increasing the descriptive power of the restriction.

Since in some systems the lexicographer may want to explicitly code in the lexicon the observation that the German reflexive pronoun in the context for Figure 6 is not translated as a reflexive pronoun in English, OLIF offers the option of a ***structural change*** statement:

```
<entry>
    <mono>
       <keyDC>
          <canForm>erinnern</canForm>
          <language>de</language>
          <ptOfSpeech>verb</ptOfSpeech>
          <subjField>general</subjField>
          <semReading>505</semReading>
       </keyDC>
    </mono>

      <transfer>
           <keyDC>
              <canForm>remember</canForm>
              <language>en</language>
              <ptOfSpeech>verb</ptOfSpeech>
              <subjField>general</subjField>
              <semReading>505</semReading>
           </keyDC>
           <trRestrictStmt>
              <trRestrict>
                   <contextStmt>
                      <context>dobj</context>
                   </contextStmt>
                   <testStmt>
                      <test>
                         <testType>datacat</testType>
                         <testDC>synType</testDC>
                         <testValue>refl-pro</testValue>
                      </test>
                   </testStmt>
              </trRestrict>
           </trRestrictStmt>

             <structChangeStmt>
                 <structChange>
                      <changeType>delInTarget</changeType>
                      <changePOS>pron</changePOS>
                 </structChange>
             </structChangeStmt>

      </transfer>
      <transfer TrDefault=”yes”>
           <keyDC>
              <canForm>remind</canForm>
              <language>en</language>
              <ptOfSpeech>verb</ptOfSpeech>
              <subjField>general</subjField>
              <semReading>505</semReading>
           </keyDC>
      </transfer>

</entry>
```

Figure 7: Structural change in transfer

Figure 7 illustrates the same approach to the analysis of structural changes that was applied to transfer restrictions in OLIF; the representation of the change was broken into the constituent parts *change type* and *part of speech of element undergoing change,* which were modeled as the OLIF data categories *changeType* and *changePOS* respectively. The resulting structural change statement includes these data categories to indicate that, in the transfer of German *erinnern* to English *remember,* the German reflexive pronoun is not transferred as a reflexive pronoun in English.

4 Streamlining OLIF entries

The XML implementation of the OLIF specification for version 2 includes a number of features that allow users to streamline their entries.

4.1 IDs

The definition of identifier (ID) attributes for targeted OLIF elements provides alternatives to repetitive listing of, for example, key data categories in cross-reference or transfer components. Users may specify either user-defined or universally-defined (GUID) identifiers as attributes to the *mono* and *keyDC* elements that may then be used to identify entries for cross-reference or transfer. Using OLIF IDs allows for a more efficient representation of the entry for English *table*, for instance:

```
<entry>
  <mono MonoUserID=0651443876>
      <keyDC>
         <canForm>table</canForm>
         <language>en</language>
         <ptOfSpeech>noun</ptOfSpeech>
         <subjField>general</subjField>
         <semReading>86</semReading>
      </keyDC>
      <monoDC>
          <monoAdmin>
             <originator>Weber</originator>
             <adminStatus>ver</adminStatus>
          </monoAdmin>
          <monoMorph>
             <inflection>like book,books</inflection>
          </monoMorph>
          <monoSyn>
             <synType>cnt</synType>
             <synFrame>[gencomp-opt]</synFrame>
         </monoSyn>
         <monoSem>
             <definition>An arrangement of words, numbers, or signs or
            combinations of them, as in parallel columns, to exhibit a set of
            facts or relations in a definite, compact, and comprehensive form.
             </definition>
             <semType>inform</semType>
         </monoSem>
       </monoDC>
    </mono>

    <crossRefer CrTarget=0591112687>
          <crLinkType>has-meronym</crLinkType>
    </crossRefer>
    <transfer TrTarget=0931445987>
    </transfer>

</entry>
```

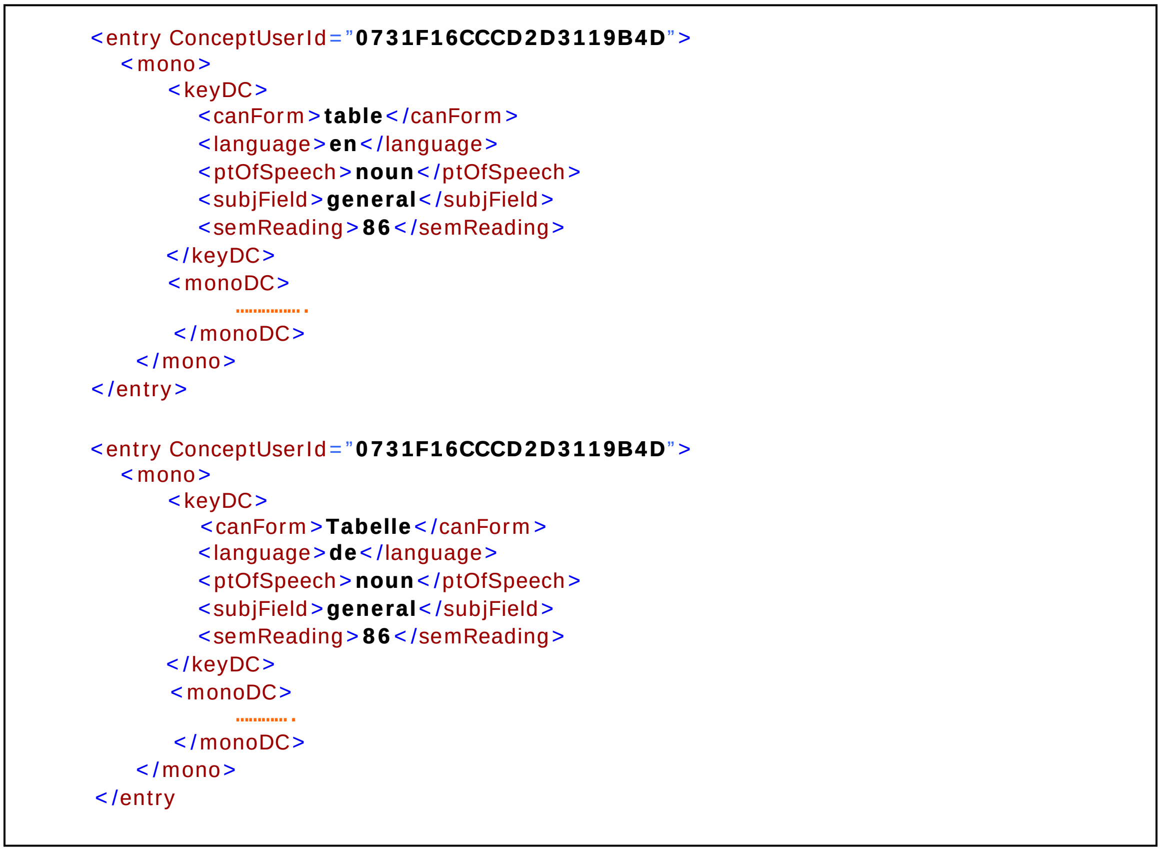
Figure 8: IDs in cross-reference and transfer

The names of the ID attributes in Figure 8 indicate that they are user-defined. Universal identifier attributes for the *mono* and *keyDC* elements (*monoUniversalID, keyDCUniversalID*), also options in OLIF, allow the user maximal interchange possibilities by referring to system-independent identifiers of entry strings.

4.2 The Concept ID

Version 2 of OLIF is designed to be flexible enough to provide different views of the data. Whereas the OLIF prototype solely supported the core OLIF model of a monolingual entry with a unidirectional transfer element, version 2 is expanded to allow as well for basic ontological modeling. With OLIF v.2, entries may be formally organized on a conceptual basis, as is the case with many terminology representation models.

Parallel with the identifier attributes for the *mono* and *keyDC* elements are the *conceptUserID* and *conceptUniveralID* attributes for the top-level element *entry.* These IDs can be used to organize entries as equivalent word senses associated with the same concepts rather than source word senses associated with transfers. Figure 9 illustrates how the English entry for *table* in Figure 2 can be remodeled with a concept ID. Rather than a single entry for *table* with a transfer component for its translation into German, there are two entries construed as equivalent via the concept ID:



```
<entry ConceptUserId="0731F16CCCD2D3119B4D">
  <mono>
     <keyDC>
       <canForm>table</canForm>
       <language>en</language>
       <ptOfSpeech>noun</ptOfSpeech>
       <subjField>general</subjField>
       <semReading>86</semReading>
     </keyDC>
     <monoDC>
          ...............
     </monoDC>
   </mono>
</entry>

<entry ConceptUserId="0731F16CCCD2D3119B4D">
  <mono>
     <keyDC>
       <canForm>Tabelle</canForm>
       <language>de</language>
       <ptOfSpeech>noun</ptOfSpeech>
       <subjField>general</subjField>
       <semReading>86</semReading>
     </keyDC>
     <monoDC>
          ............
     </monoDC>
   </mono>
</entry
```

Figure 9: Using a concept ID

With the entries for *table* and *Tabelle* related by means of a common concept ID, a bidirectional equivalence can be implied by the system that uses the OLIF data.

4.3 The OLIF header

A data file in OLIF generally follows the Terminology Markup Framework's (TMF) file structure schema of a *header* component*,* a *body* component *and* a *shared resources* component. Whereas all of the preceding description has concentrated on the body of the OLIF file, where the entries are listed, the header is an important element of the OLIF strategy. It not only provides useful global information on the data, such as the original format of the data, the owner's name and contact data, and the creator of the file, but can also indicate default values for data categories and user-specific analyses for specified data categories.

OLIF users can create a simple defaults listing in the header to avoid repetitive data coding in the entries themselves. For instance, SAP's conversion from its terminology database SAPterm to OLIF includes default values for the elements *entry status, entry source,* and *company*:

```
...
<header CreaTool="SAPterm" CreatToolVersion="46D" OrigFormat="R3 internal"
   AdminLang="DE" CreaDate="20021122112040Z" CreaId="SRINIVASANVE">
   <publStmt>
      <distributor DistributorType="cmp">
         <name>AI MLT</name>
         <telephone>06227 763321</telephone>
         <fax>06227 744119</fax>
         <eAddress EAddressType="email">v.srinivasan@sap.com</eAddress>
      </distributor>
      <owner OwnerType="natPerson">
         <name>SRINIVASANVE</name>
         <eAddress EAddressType="email">v.srinivasan@sap.com</eAddress>
      </owner>
      <availability Region="world" PubStatus="unknown" />
      <date DateValue="20021122112040Z" />
   </publStmt>
   <dataCatReg>
      <subjFieldDCS DCSType="extension">http:
         //intranet.sap.com/~sapidb/011000358700008501972002</subjFieldDCS>
   </dataCatReg>
   <contentInfo>
     <quotMarkInfo QuotMarkRet="some" QuotMarkForm="unknown" />
     <langIdUse />
     <valueDefaults>
          <valDefault ValDefaultRefName="entryStatus"
              ValDefaultRefType="el">term</valDefault>
          <valDefault ValDefaultRefName="entrySource"
              ValDefaultRefType="el">SAPterm</valDefault>
          <valDefault ValDefaultRefName="company" ValDefaultRefType="el">SAP
              AG</valDefault>
   </valueDefaults>
   </contentInfo>
</header>
```

Figure 10: Defaults and data category registry in the OLIF header

In addition, users can use the data category registry in the header to point to their own structures and values for certain data categories, including *subject field, semantic type, syntactic type,* and *inflection.* In Figure 10, the data category registry (*dataCatReg*) includes a reference to the SAP-specific subject fields for the values for that data category in the file.

5 The OLIF XSD

As noted in Section 1, the XML implementation of the OLIF v.2 specification, as it is described here, is currently in the form of a DTD. The DTD is comprised of 16 modules, approximately 130 elements and 40 attributes (see Lieske, McCormick, and Thurmair, 2002). It was designed to ensure ease of reading, maintenance and customization. As a result, OLIF instance documents can be easily processed by both humans and machines. The DTD vehicle, however, has some shortcomings, both in general and specifically for the goals of OLIF:

- Due to deviations from XML syntax, DTDs themselves are not always easily understood.
- A DTD supports a limited set of built-in data types which can be assigned only to attributes rather than elements; characteristics of the data contained in the XML instance can thus not be defined comprehensively.
- A DTD lacks support for semantic checking such as integer ranges between, for instance, 1 and 12, and durations and time spans.

While the DTD allowed the OLIF developers to implement essentially the entire OLIF v.2 specification, there were areas in the implementation which were adequate but not optimally realized due to the limitations imposed by the DTD. In reviewing alternatives, the OLIF development team decided that the XML Schema Definition Language (XSD) offered the most promising option for improving on the OLIF DTD. An XSD adheres to XML syntax, which means that it can be parsed and manipulated like any XML instance document. In addition, the XSD has more than 40 built-in data types for both attributes and elements (e.g., *dateTime*, *integer* and *boolean*) and support for checks (implemented by means of *facets* of base data types) based on patterns, number ranges or length restrictions. Provisions for object-oriented creation of data types that are user-defined allow for inheritance and re-use, and special documentation facilities provide for easy synchronization of code and comments. XSD also supports namespaces, which allows for a straightforward re-use of XML vocabularies.

Considering the advantages of XSD, the OLIF consortium recently went ahead with a development effort to implement OLIF v.2 with XSD as an alternative to the DTD that is currently available from the OLIF web site.

5.1 Custom data types

For OLIF, the main advantage of XSD is the superior data typing capability that enables powerful content-related checks. XSD supports data typing via a comprehensive list of built-in data types and a mechanism for defining custom data types. This mechanism has been liberally used in the new XSD implementation of OLIF, where, for each OLIF data category, a custom data type has been defined that specifies the possible values for the particular data category. This is a material improvement over the DTD, where, because of constraints on the formalism, values were essentially suggested. This data type is then used in the declaration of the attribute or element that formalizes the data category. Moreover, each custom data type for OLIF has been turned into a standalone schema (see Figure 11), and stored in a separate file. With this design, it is possible to include XSD representations of OLIF-specific data categories in other XML applications.

```
<!—definition of custom data type →
<xsd:schema targetNamespace=“http://www.olif.net“ xmlns=“http://www.olif.net“
xmlns:xsd=“http://www.w3.org/2001/XMLSchema“ elementFormDefault=“qualified“>
    <xsd:simpleType name=”moodType”>
        <xsd:restriction base=”xsd:string”>
            <xsd:enumeration value=”indic”>
                <xsd:annotation>
                    <xsd:documentation>indicative</xsd:documentation>
                </xsd:annotation>
            </xsd:enumeration>
            <xsd:enumeration value=”subj”>
                <xsd:annotation>
                    <xsd:documentation>subjunctive</xsd:documentation>
                </xsd:annotation>
            </xsd:enumeration>
…
        </xsd:restriction>
    </xsd:simpleType>

<!—use in declaration of element →
    <xsd:element name=”mood” type=”moodType”>
        <xsd:annotation>
            <xsd:documentation>The mood element classifies verb mood or mode.
                                    Example values: imper, cond</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:schema>
```

Figure 11: OLIF custom data type

5.2 User-extensions

From the beginning, the OLIF2 Consortium viewed OLIF as a format that should incorporate a mechanism for extensibility. In the OLIF DTD, extensibility was implemented with user-extensible parameter entities (see Figure 1212), similar to the approach of large DTDs like that for DocBook (see http://www.oasis-open.org/docbook/).

```
<!ENTITY % inflection.olif.rec.user.ext
        “(#PCDATA %inflection.user.ext;)*” >
<!—user ext. for inflection          →
<!ENTITY % inflection.user.ext “” >

<!—example for using user ext.      →
<!ENTITY % inflection.user.ext
        “| paradigm”>

<!ELEMENT paradigm (inflectedForm+)>
```

Figure 12: User extension in DTD

In the OLIF XSD, every user-extensible data category has been modeled as a union of types. As shown in Figure 13, each union includes a designated type (which follows the naming scheme *…UEType.xsd* and is stored in its own file) that captures user extensions. The extensibility is thus encapsulated in special files.

```
<xsd:schema targetNamespace=“http://www.olif.net“ xmlns=“http://www.olif.net“
xmlns:xsd=“http://www.w3.org/2001/XMLSchema“ elementFormDefault=“qualified“>
    <xsd:include schemaLocation=”inflectionENcode.xsd”/>
…
    <xsd:include schemaLocation=”inflectionUEType.xsd”/>
    <xsd:simpleType name=”inflectionType”>
        <xsd:union memberTypes=”inflectionENcodeType inflectionUEType”/>
    </xsd:simpleType>
    <xsd:element name=”inflection” type=”inflectionType”>
        <xsd:annotation>
            <xsd:documentation>The inflection element holds data about the inflection pattern(s)  of the entry
string (or its head in case of a multiword/phrasal  entry). Example use: book, 16</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:schema>

<xsd:schema targetNamespace=“http://www.olif.net“ xmlns:xsd=“http://www.w3.org/2001/XMLSchema“
xmlns=“http://www.olif.net“ elementFormDefault=“qualified“>
    <xsd:simpleType name=”inflectionUEType”>
        <xsd:restriction base=”xsd:string”>
            <xsd:enumeration value=”user extensions”>
                <xsd:annotation>
                    <xsd:documentation>this OLIF type can be extended by adding enumeration values
here.</xsd:documentation>
                </xsd:annotation>
            </xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

Figure 13: User extension in XSD

5.3 Customizing

With the OLIF XSD, additions to the possible values of a certain data category can be made by simply editing a file that is not part of the core formalization files, as in the *inflection* example in Figure 14:

```
<!—original →
<xsd:schema targetNamespace=“http://www.olif.net“ xmlns:xsd=“http://www.w3.org/2001/XMLSchema“
xmlns=“http://www.olif.net“ elementFormDefault=“qualified“>
    <xsd:simpleType name=”inflectionUEType”>
        <xsd:restriction base=”xsd:string”>
            <xsd:enumeration value=”user extensions”>
                <xsd:annotation>
                    <xsd:documentation>this OLIF type can be extended by adding enumeration values
here.</xsd:documentation>
                </xsd:annotation>
            </xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
<!—customized version →
 <xsd:schema targetNamespace=“http://www.olif.net“ xmlns:xsd=“http://www.w3.org/2001/XMLSchema“
xmlns=“http://www.olif.net“ elementFormDefault=“qualified“>
    <xsd:simpleType name=”inflectionUEType”>
        <xsd:restriction base=”xsd:string”>
            <xsd:enumeration value=”-/-n like X/Xn”>
            <xsd:enumeration value=”-/-s like Y/Ys”>
            <xsd:enumeration value=”-/-e like Z/Ze”>
                <xsd:annotation>
                    <xsd:documentation>Custom values (language en) for OLIF data category
inflection</xsd:documentation>
                </xsd:annotation>
```

```
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Figure 14: Customization of data type

5.4 Editing

The data typing capabilities of XSD allow for an easier editing of OLIF instance documents. For instance, XSD-aware structure editors can use the information on the data type and possible contents of an element to offer the permitted values for a certain element or attribute in a drop-down list:

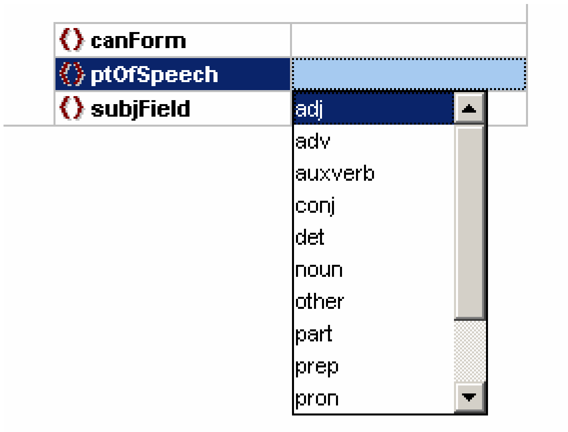


Figure 15: Editing made easy

6 Conclusions

The OLIF XSD will soon be available to users from the OLIF2 Consortium Web site (www.olif.net). The OLIF DTD has already proved itself to be a helpful tool in language data management for companies like SAP, which has long aggressively supported language technology in its translation and localization strategies. The SAP approach includes a central terminology database, MT systems for several language pairs, various translation memory (TM) tools, and a term extraction project. OLIF is being implemented at SAP for terminology exchange among its terminology database and MT system lexicons, as well as for modeling entries for term extraction. In addition, SAP is considering OLIF for representing some Web content. For users like SAP, OLIF has developed into a promising, flexible standard for lexical and terminology data exchange with broad potential for support of language applications.

References

Culum, Alexander, Christian Lieske, and Susan McCormick. "Establishing a Standard for Lexical Knowledge Exchange." Multilingual Computing and Technology. Vol.14.2 (2003): 35-42.

Computer Applications in Terminology – Data Categories (ISO 12620). International Organization for Standardization. 1999. http://www.iso.org/iso/en/. or http:www.ttt.org/clsframe/datcats.html.

Computer applications in terminology -- Machine-readable terminology interchange format (MARTIF) (ISO 12200). International Organization for Standardization. 1999. http://www.iso.org/iso/en/.

DocBook. OASIS DocBook Technical Committee. Organization for the Advancement of Structured Information Systems (OASIS). http://www.oasis-open.org/docbook/.

EAGLES. http://www.ilc.cnr.it/EAGLES.

Eurodicautom. European Commission. http://europa.eu.int/eurodicautom.

EuroWordNet. 1999. www.illc.uva.nl/EuroWordNet.

Lieske, Christian, Susan McCormick and Gregor Thurmair. "The Open Lexicon Interchange Format Comes of Age." Proceedings of the MT Summit VIII 2001.

McCord, Michael. "Using Slots and Modifiers in Logic Grammars for Natural Language." Artificial Intelligence 18 (1982): 327-367.

McCord, Michael. "Slot Grammars." Computational Linguistics 6 (1980): 31-43.

McCormick, Susan (2002). The Structure and Content of the Body of an OLIF v.2 File, OLIF2 Consortium. www.olif/net/specification/.

McCormick, Susan. "Exchanging Lexical and Terminological Data with OLIF2." Translating and the Computer 22: Proceedings of Aslib, London, 16-17 November 2000.

Open Lexicon Interchange Format (OLIF). The OLIF2 Consortium. 2002. http://www.olif.net.

SALT: Standards-based Access service to multilingual Lexicons and Terminologies. SALT Consortium; 2001. http://www.ttt.org/salt/index.html.

TermBase Exchange (TBX). http://www.lisa.org/tbx/.

Terminology Markup Framework (TMF): ISO 16642. http://www.loria.fr/projets/TMF/.