Stefanie Geldbach

Lexicon Exchange in MT
The Long Way to Standardization

Abstract
This paper discusses the question to what extent lexicon exchange in MT has been standardized during the last years. The introductory section is followed by a brief description of OLIF2, a format specifically designed for the exchange of terminological and lexicographical data (Section 2). Section 3 contains an overview of the import/export functionalities of five MT systems (Promt Expert 7.0, Systran 5.0 Professional Premium, Translate pro 8.0, LexShop 2.2, OpenLogos). This evaluation shows that despite the standardization efforts of the last years the exchange of lexicographical data between MT systems is still not a straightforward task.

1 Introduction
The creation and maintenance of MT lexicons is time-consuming and cost-intensive. Therefore, the development of standardized exchange formats has received considerable attention over the last years. On the way to standardization a number of obstacles has to be overcome (Lieske et al. 2001, Thurmair 2006):

MT developers use different data categories and values in order to represent lexicographical data. While the representation of some data categories such as gender is largely uncontroversial, much less agreement is to be found when it comes to subcategorization, semantic features or subject fields. Therefore, the development of a potential standard involves both the definition of standardized data categories and values as well as the conversion of proprietary data categories to these standards.

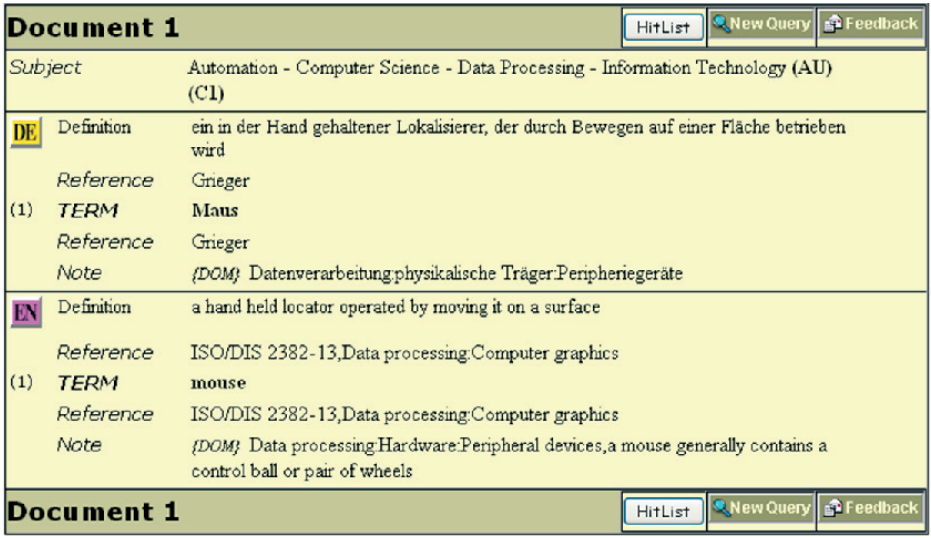
In the case of homonymy, there is possibly no one-to-one correspondence between entries in different systems. MT systems typically follow a lemma-oriented approach for the representation of homonymy which means that different semantic readings of one word are collapsed into one entry. The entry for Maus in the German monolexicon of LexShop 2.2 (see Section 3.4) illustrates this approach. This entry contains (among others) following feature-value pairs:

```
CAN “Maus”
CAT NST
ALO “Maus”
TYN (ANI C-POT)
```

The feature TYN (type of noun) which indicates the semantic type of the given noun has two values, ANI (animal) and C-POT (concrete-potent) representing two different concepts, i.e. the small rodent and the peripheral device.

Term bases usually are concept-oriented which means that different semantic readings of homonyms are stored in different entries. The definition given in the entry for Maus in the multilingual termbank EURODICAUTOM of the European Commission (see Fig. 1) which represents only one concept (here, the peripheral device) clearly illustrates this approach: If a homonymous entry such as Maus is to be imported from a lemma-oriented MT lexicon to a concept-oriented termbase the different readings of the entry have to be identified which is a non-trivial task.

2 What is OLIF2?
OLIF2 is an open XML-compliant standard specifically intended for the exchange of lexicographical and terminological data released to



Document 1		HitList / New Query / Feedback
Subject		Automation - Computer Science - Data Processing - Information Technology (AU) (C1)
DE	Definition	ein in der Hand gehaltener Lokalisierer, der durch Bewegen auf einer Fläche betrieben wird
	Reference	Grieger
(1)	*TERM*	**Maus**
	Reference	Grieger
	Note	*{DOM}* Datenverarbeitung:physikalische Träger:Peripheriegeräte
EN	Definition	a hand held locator operated by moving it on a surface
	Reference	ISO/DIS 2382-13,Data processing:Computer graphics
(1)	*TERM*	**mouse**
	Reference	ISO/DIS 2382-13,Data processing:Computer graphics
	Note	*{DOM}* Data processing:Hardware:Peripheral devices,a mouse generally contains a control ball or pair of wheels
Document 1		HitList / New Query / Feedback

Fig. 1: EURODICAUTOM entry (*http://europa.eu.int/eurodicautom/Controller*)

the public in 2002 (cf. *www.olif.net)*. OLIF2 has been developed by the OLIF Consortium, a group of major MT developers and users led by SAP[1]. Initially, OLIF was intended to facilitate the exchange of lexical data between different MT systems. OLIF2, however, aims at integrating both MT data and terminological resources by bridging the gap between the lemma-orientation of most MT lexicons and the concept-orientation of terminology management systems. "An OLIF 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" (McCormick 2002:1), which means that homonyms such as Maus or table are stored in two different entries. The body of OLIF entries contains three main data groups:

Monolingual data: each entry may contain only one monolingual group. Each OLIF entry is specified by a unique set of five data categories (*canonical form*, *language*, *part of speech*, *subject field* and *semantic reading*).

Cross-reference data define semantic relations between the given entry and other entries such as hyponymy, synonymy or meronymy.

Transfer data define the transfer relations between the given entry and other entries in different languages. Multiple transfers are possible with each transfer group representing a single, unidirectional relation.

A sample OLIF entry is shown in Figure 15[2].

3 Lexicon Exchange Functionalities in Current MT Systems

The following section contains a detailed description of the lexicon exchange functionalities of five major MT systems which is based on the information given in the respective user guides as well as the tests I conducted myself. Following systems were tested, using the language pair German – English each:

```
#format=1.0
Key     Translation     PartOfSpeech    InProp

SAP-System      SAP System      n       n
erinnern        remember        v
schämen be ashamed      v
Mangobaum       mango tree      n       m
bestehen        pass;insist;consist     v
Mutter  mother;nut      n       f
antijapanisch   anti-Japanese   a
lokalisierbar   localizable     a
Datenbankverwaltungssystem      database management system      n       n
DVS     DMS     n       n
MÜ      MT      n       f
Schweiz Switzerland     n       f
Türkei  Turkey  n       f
Mongolei        Mongolia        n       f
```

Fig. 2: Promt import format

Translate pro 8.0, a demo version is available at *http://www.lingenio.com*.

Comprendium LexShop 2.2, more information at *http://www.braintribe.com*.

OpenLogos, which can be downloaded from *http://logos-os.dfki.de/*.

For each system, it will be described how the user can create new lexicon entries and which file formats are supported for the import and export of user dictionaries. The focus is on the linguistic

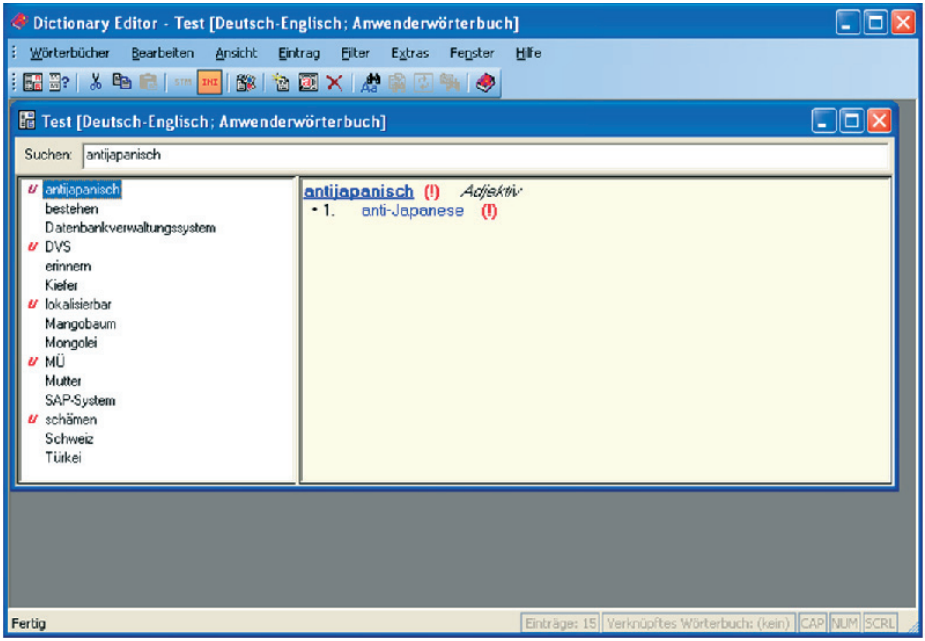


Fig. 3: @Promt Dictionary Editor

quality of the lexicographical data, i.e. the question whether the exchanged entries are complete or whether important linguistic information has to be recoded by hand. In order to test the linguistic quality the import and export test files also contained potentially difficult examples such as reflexive verbs, verbs with complex subcategorization or homonyms.

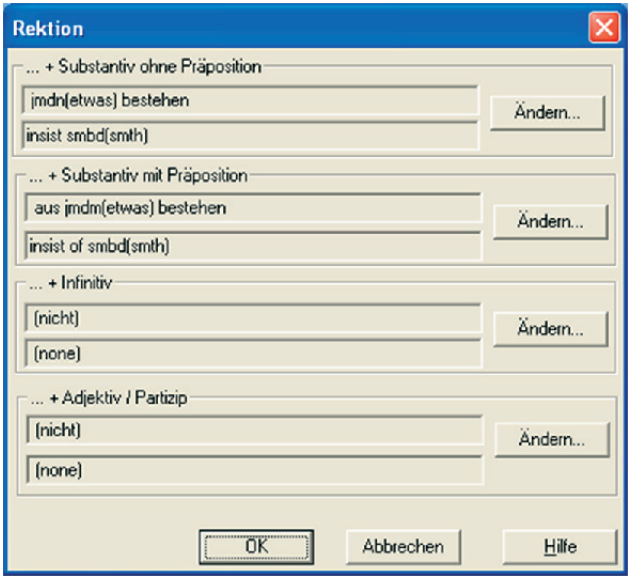


Fig. 4: SL-TL mappings in the Dictionary Editor

3.1 @Promt Expert 7.0

Promt user dictionaries are created and maintained with the help of the Dictionary Editor which guides the user through the coding process. After entering the source language word the user has to select part of speech (it is possible to code nouns, verbs, adjective and adverbs), inflection type, translation and grammatical information, notably semantic and government information. The user has the choice between two coding levels, beginner and professional. Some information such as government can only be defined at the professional level.

The location of dictionary files in the file system is controlled by Promt and not revealed to the user. A dictionary may be accessed as a file only when it is saved to a dictionary archive using the so-called Promt Backup. Dictionary archives, which are stored in a proprietary format with the extension ADC, can be used as backup copies or for copying user dictionaries to other Promt users; they cannot be imported into other MT systems, however. At present, Promt offers no possibilities of exporting user dictionaries which limits the integration of Promt user dictionaries into other MT systems.

Promt offers, however, an add-on for the automatic creation of dictionaries which enables the user to import glossaries stored as tab-delimited text files (TXT) into the user dictionary which is explained in the ADC User Guide. Import files have to be written in a specific notation which is shown in Figure 2.

The only obligatory fields are the key, i.e. the source language (SL) word, and its respective translation in the target language (TL). In order to improve the import result following fields can be added:

PartOfSpeech: the part of speech of the key. It is possible to choose between verbs (v), adjectives (a), nouns (n) and adverbs (adv).

InProp: gender and number of the SL word. Following values are possible: masculine (m), feminine (f), neuter (n), plural (pl), masculine plural (mpl), feminine plural (fpl), neuter plural (npl).

OutProp: gender and number of the TL word.
OutComment: comments and domain definitions.

Different translations of homonymous SL words, e.g. Mutter or bestehen, are separated by semicolons.

The glossary as shown in Figure 2 can be imported into an existing user dictionary using either interactive or fully automatic mode. The import result of the sample file is shown in Figure 3.

The symbol «u» marks entries which have not been verified yet, which means their grammatical information has been computed by the system and should be checked by the user. The exclamation mark (!) signals which part of the entry, i.e. SL or TL information, should be checked.

Although most of the imported entries were correct a number of problems arose which were, however, not limited to the entries marked with the symbol (!).

As the import file allows no specification of verbal subcategorization this information has to be supplied by the user. Thus, the user has to define the different syntactic frames of the verb bestehen, to map the German complements onto their English counterparts and select the respective prepositions. Here, it turned out to be impossible to encode two different prepositional government patterns for bestehen which correspond to the following readings:

(1) Der Politiker besteht auf seinem Vorschlag. '*The politician insists on his suggestion*'.
(2) Die Suppe besteht aus Wasser. *'The soup consists of water'.*

In the first example, *bestehen* governs the preposition *auf*, in the second example the prepositon *aus*. At first glance, the selection of the correct German and English prepositions does not seem to pose any problems in the *Dictionary Editor*; however, after having selected the frame for the second reading of *bestehen* as in (2) the *Dictionary Editor* changed the frame of the first reading (see Fig. 4) to *aus jmdm(etwas) bestehen / to insist of smbd(smth)* and added a further transitive frame, presumably taken from the reading *bestehen / to pass*. The information given by the user was ignored. As a result, Promt failed in disambiguating the different readings of the German sample sentences and produced the following translations for (1) and (2):

(3) The politician insists{consists} on his{its} suggestion{proposal}.
(4) The soup insists{consists} of water.

The alternative translations given here are clearly not required as the German source sentences are not ambiguous. This translation error can be explained by the assumption that the different semantic readings of the verb bestehen are internally stored in one entry in the user dictionary which in our example leads to difficulties in assigning the correct verb frames.

The representation of homonymy in the dictionary is problematic in other cases as well. Apparently, homonyms are treated as one entry in Promt dictionaries even if their gender values and inflection types are different which can be illustrated by looking at the entry for the noun Kiefer in the Promt system dictionary (see Fig. 5).

Both concepts are represented in one entry with the gender value feminine which leads to analysis and translation problems for examples such as (5) where der Kiefer is apparently analyzed as genitive NP which leads to translations such as (6).

(5) Der Kiefer ist gebrochen.
(6) Of the pine{jaw} has broken.

Consequently, the attempt to import two separate entries with different gender values for Kie-

Fig. 5: Homonymy in the Promt system dictionary

fer in the sample glossary failed because Promt automatically added the second entry (here: der Kiefer) to the first one.

3.2 Systran 5.0 Professional Premium

In Systran, the creation and maintenance of dictionary entries is handled by the SYSTRAN Dictionary Manager (SDM) which is described in detail in the Systran 5.0 User Guide (*www.systransoft.com/Support/Doc/UserGuide_EN.pdf*). SDM comes in three versions: basic, advanced and expert.

A number of features including the creation of multilingual dictionaries, import/export functionalities or the Expert Coding wizard is only provided in the expert version. The expert SDM provides three dictionary types:

User Dictionaries (UDs) which can be used to code new entries, to override target-language translations in the system dictionary and to ensure that an expression is used as a unit.

Normalization Dictionaries (NDs) which mainly serve to enhance translation consistency by normalizing SL text before or TL text after translation. NDs help, for example, to avoid orthographic variants, by ensuring that words such as *colour/color* are always spelled in the same way.

Translation Memories (TMs) which are used to store SL and TL sentence pairs. Translation memories can be built from TMX files or using Systran's Translation Project Export.

In contrast to many MT systems, the user dictionaries in Systran are not necessarily bilingual and unidirectional. It is possible to create multilingual, reversible dictionaries by including more than two languages in the user dictionary. The user is warned, however, that reversing entries in the user dictionary can have a negative impact on the translation quality.

Systran provides an easy-to-use coding interface which is meant to facilitate the integration of production-scale MT dictionaries (see Figure. 6). The only obligatory columns are source and target language(s). Systran provides multilevel coding formalisms, which range from fully automatic coding where the user only specifies SL and TL terms to expert coding:

Fully automatic coding: SDM automatically analyzes and codes the entry. The user does not have to specify any information except the SL and TL language columns although it is advisable to select the appropriate category (proper noun, adjective, verb, adverb, preposition, sequence, acronym) oneself as the au-

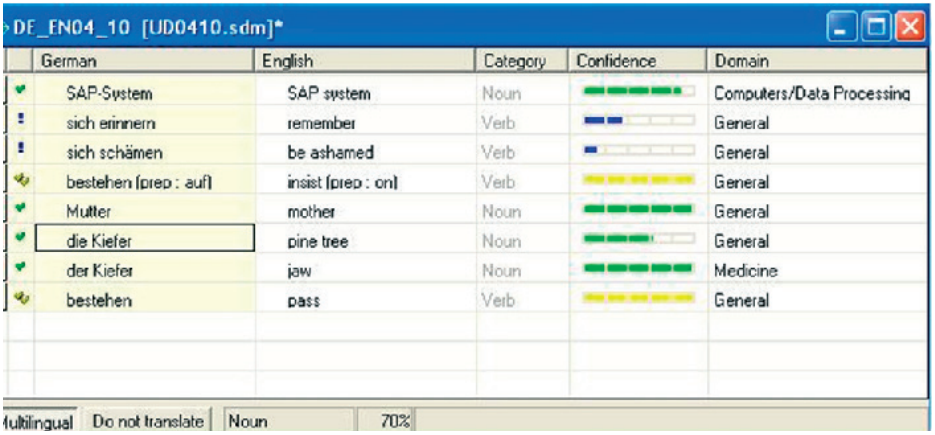


Fig. 6: SDM coding interface

tomatic analysis may lead to wrong category assignments.

Intuitive Coding: Systran's Intuitive Coding technology (Senellart et al. 2003) enables the conversion of simple user dictionaries into the knowledge representation of the MT dictionary. The coding engine converts various clues supplied by the user into linguistic information. It is possible, for example, to use particles or determiners in the entries in order to determine the grammatical category, thereby avoiding ambiguities existing between different categories in case of homonymy (see Table 1).

English	German
to light	anzünden
a light	Licht
light	leicht

Table 1: Systran Intuitive Coding

Expert coding: The coding wizard which is provided in the expert SDM allows the complete modification of Systran's analysis of the entry. Using expert coding (see Figure 7) it is possible to code detailed morphological, syntactic, semantic and typographical information by hand.

The confidence level of the entries is indicated in a confidence column on the left side of the SDM coding interface. A single checkmark in the status column next to the entry indicates a satisfactory definition. Double checkmarks indicate that the entry has been validated, e.g. by using expert coding. Exclamation marks appear when a warning has been issued; here, the entry should be reviewed.

The Dictionary Manager also provides import and export features which are described in Appendix D of the Systran User Guide. It is possible to open dictionaries created with a spreadsheet application such as Microsoft Excel or tab-delimited text files. The dictionaries have to be specifically formatted before they can be imported into SDM. Text files to be imported into SDM have to contain dictionary content and document headers which are listed in Table 2. The sample text file given in Figure 8 is formatted for importing into SDM. Additionally, the SDM Import Menu lists the possibility to import TMX and XML files. In the respective section of the online Systran 5.0 User Guide, however, the import of XML files is not mentioned at all so that users have to find out for themselves for which other applications these files are intended. Attempts to import Translate pro XML files (see

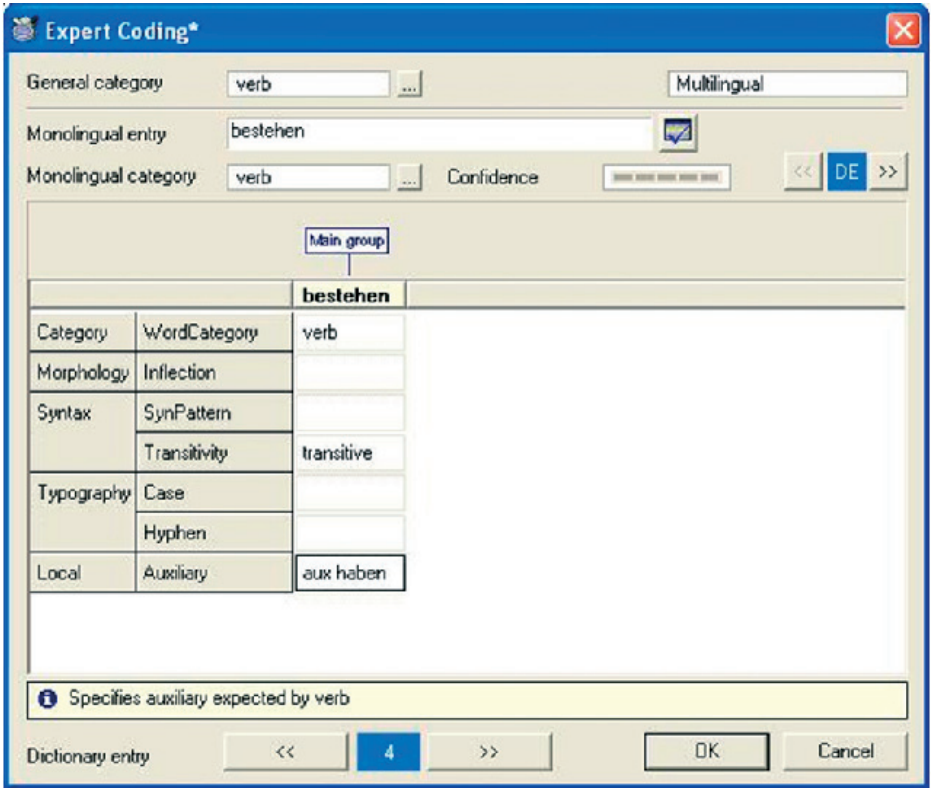


Fig. 7: Systran Expert Coding Wizard

```
#ENCODING=UTF-8
#COVERED DOMAINS=Medicine,Computers/Data Processing
#PRIORITY=1
#SUMMARY=Test
#MULTI
#DE      EN       NOTE     DOMAINS HEADWORD_DE      HEADWORD_EN
die Kiefer       pine tree
der Kiefer (noun)        jaw      Medicine
Datenbankverwaltungssystem       database management system       ?
SAP-System       SAP system       Computers/Data Processing
bestehen         pass
Mutter  mother
lokalisierbar    localizable      Computers/Data Processing
#DNT
#DE      NOTE     DOMAINS
```

Fig. 8: Systran TXT import

Header	Description of Input
#AUTHOR=	Optional: contains the name of the creator of the dictionary
#EMAIL=	Optional: email address of the creator of the dictionary
#COVERED DOMAINS	Optional: lists all domains
#GENERAL DICTIONARY DOMAINS	Optional: lists the system domains
#MULTI	Required: determines the UD tab Multilingual for the header information that follows
#SUMMARY=	Required: the name of the UD file
#<Languages> <Informational Columns> =	Required: designates all informational columns for the UD
#DNT	Required: determines the UD tab Do not translate for the inormation that follows

Table 2: TXT import in Systran

Section 3.3) failed, whereas the import of Multiterm iX XML files was successful.

Just as tab-delimited text files and Microsoft Excel files can be imported into SDM, user dictionaries created in SDM can be exported to these formats.

Although Systran developers are working on OLIF2 support, this format has not been integrated in any commercial product yet[3].

3.3 Translate pro 8.0

The translation system Translate pro 8.0 from the Heidelberg company Lingenio shares a common history with the Personal Translator from the Munich-based company Linguatec. Both systems originate from LMT, a machine translation system initially developed by IBM (McCord 1989). Until 2004, the MT system was developed exclusively in Heidelberg and distributed by Linguatec in Munich. After the restructuring of Linguatec Entwicklung & Services in 2004, the Heidelberg developer team founded Lingenio and launched their MT system under the name Translate pro. Because of the common ancestry of the two systems it is possible to copy proprietary user dictionaries created in Translate pro directly into the Personal Translator 2001 – 2004 and vice versa.

The lexicon exchange with other MT systems is not as straightforward, though. Similar to other systems,

Translate pro offers the possibility to import bilingual glossaries as text files. As these glossaries contain only word pairs and no information on the grammatical category the user is advised to include in one import file only words belonging to the same part of speech, e.g. nouns or verbs or adjectives. The TXT file contains only word pairs, e.g.:

```
Kiefer@@@pine tree
Mangobaum@@@mango tree
Datenbankverwaltungssystem@@@da
tabase management system
```

Apart from TXT files it is also possible to import and export XML dictionaries.

The drawbacks of the Translate pro XML entry structure are illustrated by looking at the entry for the reflexive verb sich schämen 'to be ashamed' which was coded in a new user dictionary. This verb has different syntactic frames including an optional genitive object as in (7)

(7) Er schämte sich seines Verhaltens. *'He was ashamed of his behaviour'.*

This subcategorization which actually has not been considered in the current system dictionary can easily be coded in a user dictionary. The

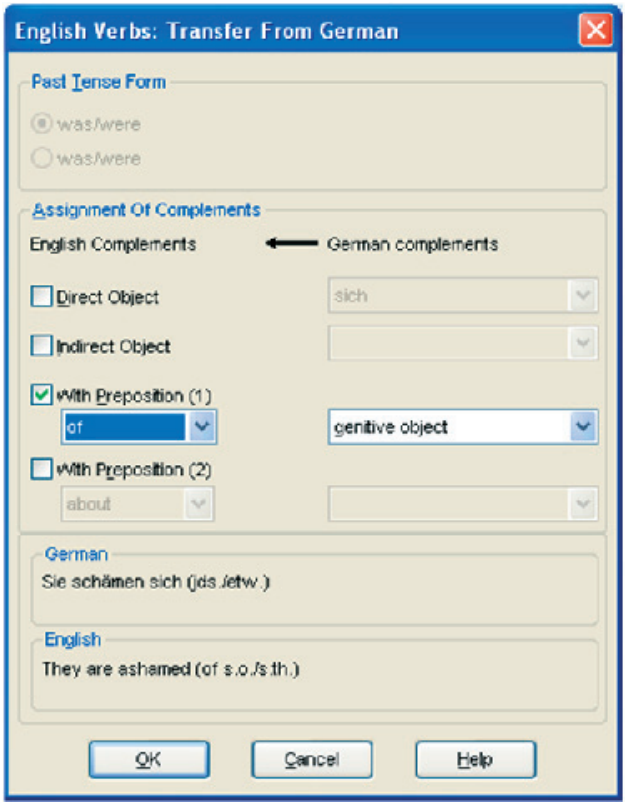


Fig.9 : SL-TL assignments in Translate pro

user has to select the respective German complements by activating Extended Coding and to map them onto their matching English counterparts.

The German reflexive pronoun sich has to be deleted, i.e. it is assigned no English complement while the German genitive object is mapped onto an English prepositional object with the preposition of. The assignment of complements is shown in Figure. 9. These assignments are imperative for producing a correct translation and should therefore be preserved during lexicon export. The exported entry for sich schämen, which illustrates the XML structure used in Translate pro, contains following information:

```
<entry>
<hdterm>schämen
</hdterm>
<hom>
<epos>v</epos>
<sense>
<edef>Er schämte
sich seines
Verhaltens.</edef>
<target>
<trans>be ashamed</
trans>
<tpos>v</tpos>
</target>
</sense>
</hom>
</entry>
```

The tag `<edef>` is optional, all other tags are obligatory. It is obvious that this XML format contains no tags which correspond to `<synFrame>` in the mono section or `<structChangeStmt>` in the transfer section of an OLIF2 entry. Therefore, the information on the German subcategorization and the structural changes during transfer is lost during lexicon export. The sample transfer for German sich erinnern to English remember which is included in (McCormick et al. 2004) shows exactly how structural changes such as the deletion of a German reflexive pronoun would have to be coded in OLIF2:

```
<structChangeStmt>
<structChange>
<changeType>delInTarget
</changeType>
```

```
<changePOS>pron</changePos>
</structChange>
</structChangeStmt>
```

The structural change is represented in the OLIF data categories changeType and changePOS.

The representation of homonymy in the Lingenio XML structure is also an interesting case which will again be illustrated with sample entries for German Kiefer and the English translations jaw and pine tree. Basically, two XML notations are possible to code the two English translations. In the first notation, both translations are included in one entry with two target groups:

```
<entry>
...
<target>
<trans>jaw</trans>
<tpos>n</tpos>
</target>
<target>
<trans>pine tree</trans>
<tpos>n</tpos>
</target>
...
</entry>
```

As a result the import function generates only one noun entry with two translations which means that only one gender value, i.e. either feminine or masculine can be selected.

The second possibility consists of coding two distinct entries in the XML file with one `<target>` group each. This solution, which results in creating two noun entries during lexicon import (see Figure 10) is clearly preferable. Although the first noun has wrongly been assigned masculine gender by the Translate pro import function the user can at least correct the incorrect gender and create two noun entries for Kiefer with different gender values.

The XML entries which are generated by the export function are intended for importing Translate pro user dictionaries into other applications. Unfortunately, the documentation does not mention which applications apart from the Personal Translator actually support the XML format described here. Attempts to import Translate pro XML files into Systran Professional Premium and Multiterm iX failed both.

3.4 Comprendium Translator – LexShop 2.2

LexShop is a sophisticated tool for the creation and maintenance of Comprendium-style dictionaries developed by Braintribe lingua. Braintribe lingua offers a wide range of home and enterprise translation solutions which evolved from the former METAL technology. Home desktop products include the machine translation system T1 which is distributed by Langenscheidt. LexShop is included in Comprendium Lexicographer, a package addressed to professional corporate and academic users which consists also of a Translator Engine and a Translator Desktop.

Lexicographers using LexShop have full access to the internal lexicon structure which sig-

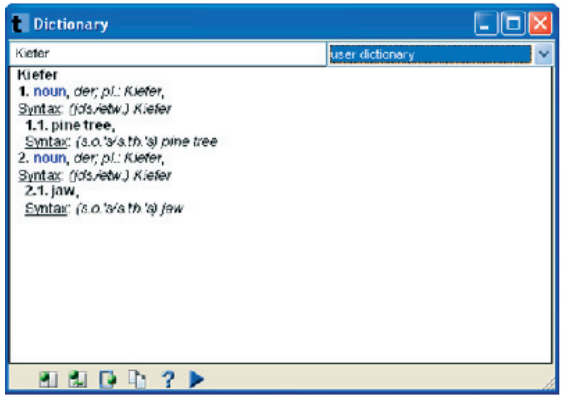


Fig. 10: Homonymy import in Translate pro

nificantly enhances their control over the coding process. They are given elaborate coding options equivalent to those of system developers which, however, presupposes an in-depth understanding of the translation process and the system architecture as a whole which is outlined in the documentation: Comprendium is a typical transfer system with a modular system architecture, i.e., the translation process can be divided into analysis, transfer and generation. The system consists of three main components: the software kernel which directly controls the translation process and invokes the different linguistic modules, the lingware which contains the grammatical rules and procedures required for analysis, transfer and generation and the lexicons. The system requires two kinds of lexicons, monolingual lexicons (monolexicons) which are used during analysis and generation and bilingual transfer lexicons which map SL words or phrases onto their TL equivalents. All lexicographical information is featurized, i.e. stored as feature-value pairs (FVPs) with approximately 100 different features being used in the MT lexicons. In LexShop, lexicon entries have to be coded for each lexicon separately, i.e. source monolexicon, target monolexicon and transfer lexicon. The coding of a new translation, e.g. from German Lokalisiererin to English localizer involves several steps:

Creation of German monolingual entry: The lexicographer has to check whether the German monolexicon already contains the entry *Lokalisiererin*. If not, a new entry has to be created.

Creation of transfer entry: The user has to create a transfer entry in the German-English transfer dictionary which contains the required translation from German *Lokalisierin* to English *localizer*.

Creation of English monolingual entry: In the last step, the entry *localizer* has to be added to the English monolexicon. In this case, the English monolexicon already contained an entry for *localizer* whose values for the features TYN (type of noun) and SX (sex) had to be modified.

LexShop supports the development of a lexicon by offering default values for mono and transfer features. When coding a monolingual entry the lexicographer only has to select the canonical form (CAN) and the category (CAT). All other obligatory values are automatically computed by the system. Following FVPs are contained in the German entry Lokalisiererin (see Figure 11):

ALO (allomorph): The ALO value is the string to which inflectional endings are attached. A canonical form (CAN) can have several allomorphs, e.g. the German verb *bringen* has three different ALO values, *bring*, *brach*, and *bräch*.

CL (morphological class): The CL feature describes the inflection, i.e. which nominal flexes are used in the singular and plural.

GD (gender): The GD value of the given canonical form, in this case feminine.

KN (kind of noun): KN is a syntactic-semantic feature which is used to distinguish between mass and count nouns. *Lokalisiererin* takes the value CNT, i.e. this noun is countable.

SX (sex): This feature indicates the natural gender of the given noun.

TYN (type of noun): The feature TYN indicates the semantic type of the given noun and is used, for example, in order to code selectional restrictions in syntactic frames. LexShop uses a list of 20 values for TYN. *Lokalisiererin* has the value HUM (human being).

The lexicographer can add further values to the entry or modify values which were defaulted by the system.

LexShop also provides quite elaborate import and export functionalities. Monolingual entries

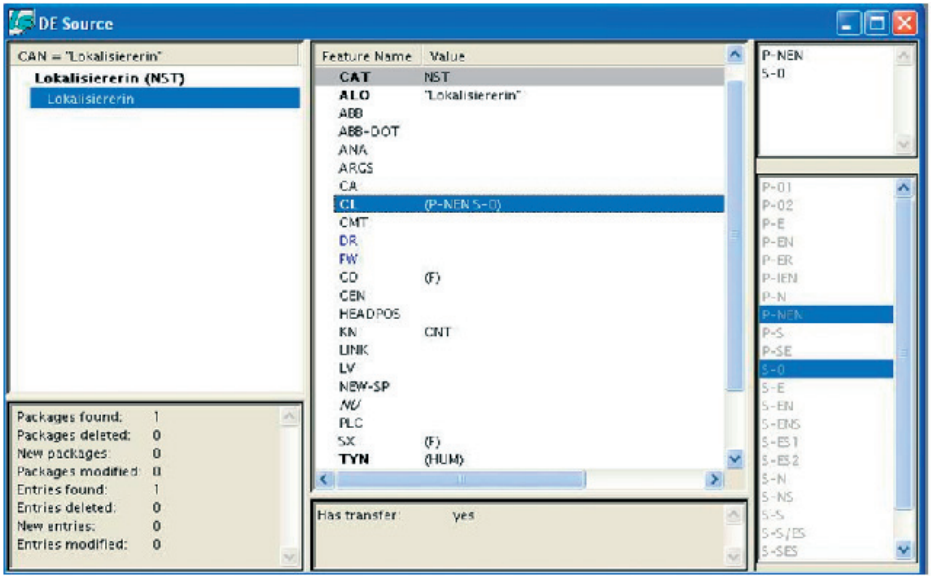


Fig. 11: Entry in the German monolexicon

(monopackages) can be imported as CSV lists and in LIF (lexicon internal format). The formats supported for importing transfer entries are LIF, CSV and IMP, an encrypted format of exported Comprendium transfer entries. LIF is a proprietary format which contains all feature value pairs of the entry in the internal notation, e.g.:

```
:LANGUAGE DE
:FORMAT INTERNAL
(CAN "Brief" CAT NST ALO
"BRIEF" CL (P-E S-S/ES) GD (M)
KN CNT SX (N) TYN (ABS CNC
SEM))
```

CSV lists allow to import one lexicon per file, i.e. either mono or transfer entries.

CSV import ranges from very basic to highly complex entries. The only features which always have to be given are CAN and CAT, missing obligatory features are defaulted. The sample import file shown in Figure 12 contains the additional features ALO, GD, KN, SX, ABB (abbreviation), TYN and ARGS (arguments). If the import file contains FVPs which are not defined in the lexicon specification LexShop displays an error message.

The CSV file for the corresponding transfer entries is shown in Figure 13. Each entry contains SL and TL canonical form and category (SLCAN, SLCAT, TLCAN, TLCAT). The TAG feature denotes the subject area the transfer entry belongs to, e.g. GV (general vocabulary).

LexShop displays the imported entries in temporary windows according to whether they are new or conflicting entries, that is entries with CAN and CAT values which already exist in the lexicon. In the sample import file, all entries except antijapanisch already existed in the German monolexicon. By displaying the corresponding mono entries the lexicographer can easily compare the new entries with the existing ones and decide which entries he wants to keep or discard (see Figure 14). Additionally, LexShop checks

```
LANGUAGE;DE;;;;;;;
FORMAT;CSV;;;;;;;


CAN;CAT;ALO;GD;KN;SX;ABB;TYN;ARGS

Kiefer;NST;Kiefer;(M);CNT;(N);;(BPART);
Kiefer;NST;Kiefer;(F);CNT;(N);;(PLANT);
lokalisierbar;AST;lokalisierbar;;;;;;;
antijapanisch;AST;antijapanisch;;;;;;;
DVS;NST;DVS;(N);;(N);T;(C-SEM);
lokalisieren;VST;lokalisier;;;;;;(([$SUBJ N1) OPT($DOBJ N1))]
```

Fig. 12: CSV import of German mono entries

whether the imported entries were syntactically correct.

The strength of the exchange formats used in LexShop lies in the complete representation of the lexicon features. It is possible to export and import complete entries with all FVPs, thus preserving the complete lexicographical information coded. This advantage becomes obvious when comparing the import structure for verb entries in different MT systems. In Translate pro, for example, the information on the German syntactic frame and necessary structural changes from German to English was lost in the exported entry for sich schämen. In LexShop, this type of syntactic information can be specified with the help of the features ARGS (arguments) in import/export mono files (cf. the entry for lokalisieren in Fig. 12) and XFMS (structural transformations to be performed during transfer) in the import/export transfer files.

All user-modified entries can be exported from LexShop. At present, the only export formats which are supported by LexShop are LIF

```
LANGUAGE;DE_EN;;;
FORMAT;CSV;;;


SLCAN;SLCAT;TLCAN;TLCAT;TAG

Kiefer;NST;pine tree;NST;(GV)
Kiefer;NST;jaw;NST;(GV MED)
lokalisierbar;AST;localizable;AST;(DP)
antijapanisch;AST;anti-Japanese;AST;(GV)
DVS;NST;DMS;NST;(DP)
lokalisieren;VST;localize;VST;(DP)
```

Fig. 13: CSV import of German-English transfer entries

for monopackages and LIF and IMP for transfer entries. However, Braintribe developers are currently working on import converters for OLIF and MARTIF and export converters for CSV and OLIF[4].

OLIF2 is already supported by the Braintribe terminology extraction tools TermExtract and BiExtract. TermExtract is a tool for monolingual term extraction which takes text files as input and produces HTML or OLIF files as output. The resulting monolingual OLIF entries include the key data categories as well as administrative information and an example which illustrates the context the given term occurred in (see Figure 15). BiExtract is a tool for the extraction of bilingual glossaries from translation memories. The input to BiExtract is a translation memory for a given language pair and a file (TXT or OLIF) containing terms in the source language. The results are given in HTML files.

3.5 OpenLogos

Logos is one of the veteran MT systems whose history reaches back to 1970 when US government agencies were in need of a English-Vietnamese translation system which triggered the development of the Logos system (Scott 2003). In 2001, Logos Corporation transferred its technology to the German company GlobalWare which announced the release of Logos as open source in cooperation with the Saarbrücken-based DFKI in September 2005. In the future, anyone can test and use Logos or develop new components for additional language pairs. OpenLogos (or *LogOSMaTran*), the open source version of the Logos system for Linux is available at *http://logos-os.dfki.de/*. GlobalWare is currently also working on a Web-based test drive of the Language Deve-

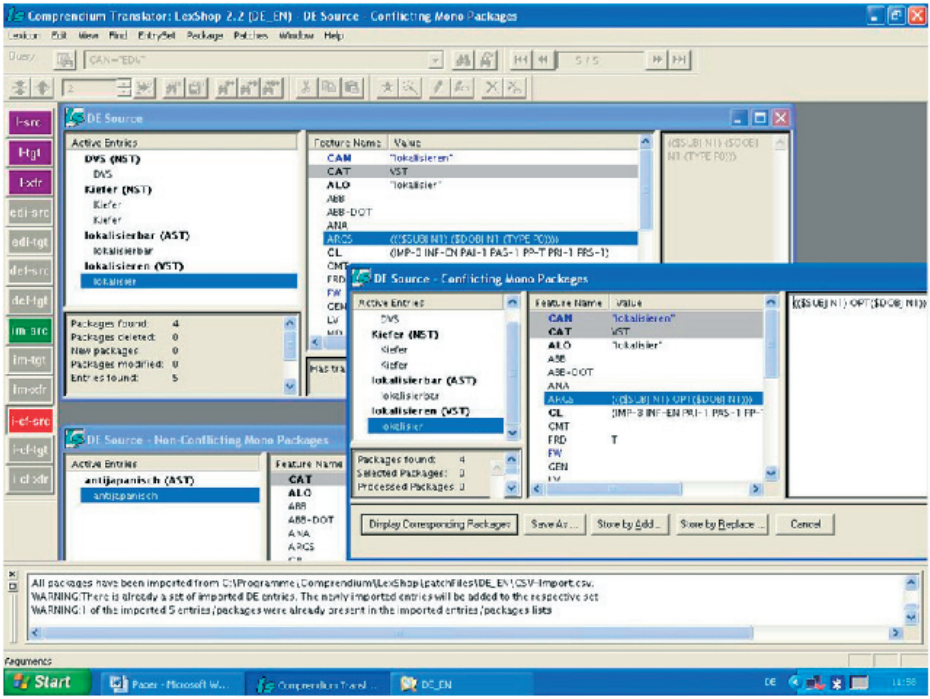


Fig. 14: Importing conflicting monopackages in LexShop

lopment Environment (LDE) of the LogOSMa-Tran engine at *http://www.logos-mt.com*.

The creation of new dictionary entries in the standalone OpenLogos version is handled by the TermBuilder (see Figure 16). The coding process is further supported by the so-called SAL wizard (SAL stands for semantico-syntactic abstraction language, i.e. the linguistic representation language used in Logos) which autocodes part of the lexicon features.

Logos also supports several file formats for the import/export of dictionary entries[5]. The format of an import file is either OLIF, TXT or TermSearch, a proprietary format. The format of an export file is either TXT, XML or OLIF. Text files used for lexicon exchange have to contain following fields:

```
Source_Language; Source_
Word; Head_Word; Source_POS;
Source_Gender; Target_Language;
Target_Word; Target_Gender;
Company_Code; Subject_Matter_
Code; Lexicon_Source.
```

All fields have to be separated by semicolons, e.g.:

```
DE;Mangobaum;Mangobaum;Noun;Mas
c;EN;mango tree;;LOG;001;null
```

In this example, which is taken from an export file, the value for the target gender is empty.

The XML format used in Logos is actually more or less identical to the XML files used

```
<entry>
- <mono>
  - <keyDC>
      <canForm>hatching egg</canForm>
      <language>EN</language>
      <ptOfSpeech>common noun</ptOfSpeech>
      <subjField />
    </keyDC>
  - <monoDC>
    - <monoAdmin>
        <entryStatus>term</entryStatus>
      </monoAdmin>
    </monoDC>
  - <generalDC>
      <example>The measures specifies that no live poultry and<t> hatching eggs</t> may
        be transported within the Netherlands nor dispatched from the<font
        color=red> ...</font></example>
      <note>frequency: 4</note>
    </generalDC>
  </mono>
</entry>
```

Fig. 15: OLIF entry generated by TermExtract

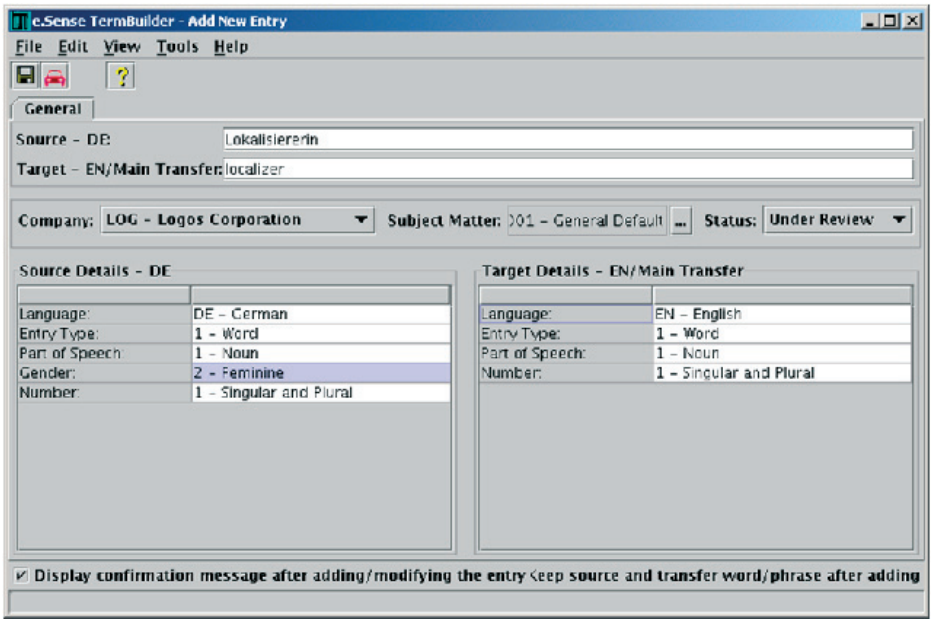


Fig. 16: OpenLogos TermBuilder

by Translate pro (see Section 3.3) and Linguatec. The exported entry for the entry Mangobaum, for example, has the following structure:

```
<entry>
<hdterm>Mangobaum</hdterm>
<hom>
<epos>n</epos>
<sense>
<target>
<trans>mango tree</trans>
<tsubjcode>001</tsubjcode>
</target>
</sense>
</hom>
</entry>
```

Unfortunately, Logos currently does not support OLIF2 as an export format but only an older version of OLIF which was developed in the OTELO project, an earlier standardization initiative (see Figure 17).

Conclusion

The evaluation in this paper has shown that the lexicon import/export functionalities actually supported by major MT systems are still only partially compatible which complicates the exchange of user dictionaries as part of the lexicographical information may have to be recoded. Despite the efforts of the OLIF Consortium to streamline the exchange of lexicographical data many MT vendors still do not support OLIF2. In order to facilitate the integration of OLIF functionalities into other programs the OLIF Consortium has developed a number of tools such as a CSV-to-OLIF converter which can be downloaded from the OLIF website. As OLIF2 is also intended for the integration of terminological data further acceptance of this format will-depend on the support of OLIF2 in other CAT tools such as termbases or terminology extraction systems.

```
<OLIF>
<HEADER>
<AUTHOR = logos>
<DATE = 2005-21-12>
<CHARACTER CODE = ISO LATIN 8859/1>
<PROJECT = mt>
<SOURCE = logos>
<TARGET = otelo>
</HEADER>
<BODY>
<entry>
<mono>
<canForm = Mangobaum>
<ptOfSpeech = noun>
<subjField = LOG-001>
<language = ger>
<entryType = cmp>
<TSTAT = mt>
<originator = logos>
<CE-DATE = 2005-20-12>
<updater = logos>
<modDate = 2005-20-12>
<SEMT = (cnc,cnt,plant)>
<gender = (m)>
<synType = (cnt)>
<USE = offline>
</mono>
<transfer>
<canForm = mango tree>
<ptOfSpeech = noun>
<subjField = LOG-001>
<language = eng>
<EQ = sub>
<X-SRC = logos>
```

Fig. 17: OLIF Export in Logos

Acknowledgments

I wish to thank all people who contributed to this paper either by providing demo versions or by answering my questions to their systems, including Julia Epiphantseva, Walter Kasper, Tamara Kotek, Stephan Küpper, Jean Senellart and Maria Strobel.

References

Lieske C., McCormick S., Thurmair G. (2001). "The Open Lexicon Interchange Format (OLIF) comes of Age". In: Proceedings of MT Summit VIII, Santiago.

McCord M. (1989). "Design of LMT: A Prolog-Based Machine Translation System". In: Computational Linguistics 15(1), pp. 33-52.

McCormick, S. (2002). "The Structure and Content of the Body of an OLIF v.2.0/v.2.1 File, OLIF2 Consortium", *http://www.olif.net*.

McCormick, S., Lieske C., Culum A. (2004). "OLIF v.2: A Flexible Language Data Standard", *http://www.olif.net*.

Scott, B. (2003). "The Logos Model: An Historical Perspective". In: Machine Translation 18, pp. 1-72.

Senellart J.,Yang J., Rebollo A. (2003). "SYSTRAN Intuitive Coding Technology". In: Proceedings of MT Summit IX, New Orleans.

Thurmair, G. (2006). "Exchange Formats: TBX, OLIF and Beyond". In: Geldbach, St., Seewald-Heeg U. (eds.): "Exchange of Lexical and Terminological Resources", LDV-Forum 21(1), pp 43-55.

Endnotes

1 To my knowledge, all of the MT developers (with the exception of Promt) mentioned in this paper were (or are) members of the OLIF Consortium.

2 For a discussion of OLIF2 and further sample entries see also the contribution by Thurmair in this volume.

3 Jean Senellart, personal communication. Senellart also reports on difficulties concerning the representation of multiwords such as *voiture de course rapide* in OLIF2. For MT processing, it is necessary to include the information that the adjective *rapide* agrees with *voiture* and not with *course* which cannot be stated explicitly in an OLIF entry.

4 Tamara Kotek, personal communication.

5 Due to technical problems, I could not test the LogOSMaTran LDE at the respective website. I am therefore indebted to Walter Kasper (DFKI) for providing me with sample import and export files generated by Logos.