Band 21 – Heft 1 – Jahrgang 2006 – ISSN 0175-1336

Zeitschrift für Computerlinguistik und Sprachtechnologie
GLDV-Journal for Computational Linguistics and Language Technology

LDV Forum

Exchange of Lexical and Terminological Resources

Beiträge des Workshops des
GLDV-Arbeitskreises Maschinelle Übersetzung
Köthen, Juni 2005

Herausgegeben von
Stefanie Geldbach und Uta Seewald-Heeg

Gesellschaft für linguistische Datenverarbeitung www.gldv.org

Exchange of Lexical and Terminological Resources

Austausch von Terminologie zwischen Systemen zur Terminologieverwaltung, computerunterstützten und maschinellen Übersetzung

Impressum

LDV-Forum
ISSN 0175-1336

Zeitschrift für Computerlinguistik und Sprachtechnologie
GLDV-Journal for Computational Linguistics and Language Technology – Offizielles Organ der GLDV

Herausgeber

Gesellschaft für Linguistische Datenverarbeitung e. V. (GLDV)
Juniorprofessor Dr. Alexander Mehler, Universität Bielefeld,
alexander.mehler@uni-bielefeld.de
Prof. Dr. Christian Wolff, Universität Regensburg
christian.wolff@sprachlit.uni-regensburg.de

Band 21 – 2006 – Heft 1
Herausgeber

Themenheft Terminologieaustausch
Prof. Dr. Uta Seewald-Heeg, Hochschule Anhalt und Dr. Stefanie Geldbach

Anschrift der Redaktion

Prof. Dr. Christian Wolff
Universität Regensburg
Institut für Medien-, Informations- und Kulturwissenschaft
D-93040 Regensburg

Wissenschaftlicher Beirat

Vorstand, Beirat und Arbeitskreisleiter der GLDV
http://www.gldv.org/cms/vorstand.php, http://www.gldv.org/cms/topics.php

Erscheinungsweise

Zwei Hefte im Jahr, halbjährlich zum 31. Mai und 31. Oktober. Preprints und redaktionelle Planungen sind über die Website der GLDV einsehbar (*http://www.gldv.org*).

Einreichung von Beiträgen

Unaufgefordert eingesandte Fachbeiträge werden vor Veröffentlichung von mindestens zwei ReferentInnen begutachtet. Manuskripte sollten deshalb möglichst frühzeitig eingereicht werden und bei Annahme zur Veröffentlichung in jedem Fall elektronisch und zusätzlich auf Papier übermittelt werden. Die namentlich gezeichneten Beiträge geben ausschließlich die Meinung der AutorInnen wieder. Einreichungen sind an die Herausgeber zu übermitteln.

Bezugsbedingungen

Für Mitglieder der GLDV ist der Bezugspreis des LDV-Forums im Jahresbeitrag mit eingeschlossen. Jahresabonnements können zum Preis von 25,- € (inkl. Versand), Einzelexemplare zum Preis von 15,- € (zzgl. Versandkosten) bei der Redaktion bestellt werden.

Satz und Druck

Uta Seewald-Heeg und Christian Wolff mit *Adobe InDesign CS 3.0.1,*
Druck: Druck TEAM KG, Regensburg

Stefanie Geldbach und Uta Seewald-Heeg

Editorial

Liebe GLDV-Mitglieder, liebe Leserinnen und Leser des LDV-Forums,

Heft 1/2006 des LDV-Forums erscheint erneut als Sonderheft, das Beiträge des im Juni 2005 an der Hochschule Anhalt (FH) in Köthen durchgeführten Workshops des Arbeitskreises „Maschinelle Übersetzung“ der GLDV zum Thema „Austausch von Terminologie zwischen Systemen zur Terminologie-verwaltung (TVS), computerunterstützten (CAT) und maschinellen Übersetzung (MÜ) / Exchange of Lexical and Terminological Resources in Machine Translation (MT), Computer-Aided Translation (CAT) and Terminology Management Systems (TMS)” enthält.

Der Workshop erfreute sich reger Nachfrage, da Fragen des Austauschs von Terminologie in nahezu allen Bereichen der Anwendung von Sprachtechnologie von entscheidender Bedeutung für eine reibungslose Abwicklung von Projekten sind.

Aufgrund der internationalen Teilnehmerschaft der Veranstaltung hatten sich alle Vortragenden bereit erklärt, ihre Vorträge in englischer Sprache zu halten. Aus diesem Grunde liegen auch die schriftlichen Fassungen dieser Beiträge in englischer Sprache vor.

Sowohl die Präsentationen der in diesen Band eingeflossenen Beiträge als auch einzelne Tagungs-beiträge, die nicht in der vorliegenden Ausgabe vertreten sind, können auf der Webseite des Workshops unter *http://www-koethen.heeg.de/GLDV2005/programm.htm* eingesehen werden.

Köthen und Heidelberg, im Februar 2006

Stefanie Geldbach und Uta Seewald-Heeg.

Inhaltsverzeichnis

LDV FORUM – Band 21(1) – 2006
Themenheft Exchange of Lexical and Terminological Resources

Uta Seewlad-Heeg, Stefanie Geldbach

Introduction

The widespread use of Computer Aided Translation (CAT) tools has revolutionized the daily work of translators and localizers. In an increasingly automated workflow the use of standardized formats provides a significant contribution to the management and quality assurance in large translation projects. Consequently, the Localization Industry Standards Association (LISA) is actively promoting the development of various standards covering the different stages of the translation workflow from job creation to archival (see Figure 1)[1]. In this scenario, translation memories, term bases and machine translation (MT) lexicons are regarded as linguistic assets. Standards provide a way to protect these assets against market and technology changes since they keep users from being locked into a particular CAT tool.

While TMX (**T**ranslation **M**emory e**X**change format) which was defined first in 1998 (*www.lisa.org/standards/tmx/*) has been widely adopted as standard exchange format and is nowadays supported not only by most translation memory systems but also by a growing number of MT vendors such as Systran, linguatec, Lingenio or braintribe, the picture is less clear on the terminological and lexical side. Although various standards such as OLIF or TBX have been proposed for the exchange of terminological and lexical resources many vendors of MT or other CAT tools have not yet adopted these standards and continue to use proprietary formats only.

This LDV Forum volume contains the proceedings of an international workshop of the GLDV interest group “Machine Translation” entitled “Exchange of Lexical and Terminological Resources in Machine Translation (MT), Computer-Aided Translation (CAT) and Terminology Management Systems (TMS)” which was held at Anhalt University of Applied Sciences (Hochschule Anhalt) in Köthen (Anhalt) (*http://www.inf.hs-anhalt.de*) on June 17, 2005. The workshop brought together MT developers, researchers and translators interested in the integration of lexical and terminological resources. Consequently, the issues addressed at the workshop ranged from practical problems arising during the mass export/import of terminology to fundamental conceptual differences between term bases and MT lexicons which complicate standardization.

Uta Seewald-Heeg reports on the exchange functionalities of terminology management systems (TMS) such as MultiTerm, TermStar, or SDLTermBase (to name just a few) focussing on the question whether the formats currently supported by these systems enable terminology exchange without loss of information. Wolfgang Zenk’s contribution centres on the UniTerm TMS which is developed by Acolada GmbH. Zenk discusses the database design and the various import/export formats currently used by UniTerm and elaborates on the problems of blind terminology interchange.

Stefanie Geldbach gives an overview of lexicon exchange formats currently used by commercial MT systems. Her paper also investigates whether the standardization efforts of the OLIF Consortium have actually resulted in a widespread acceptance of the OLIF2 standard.

Gregor Thurmair discusses two of the formats which are promoted by LISA, namely TBX (Term Base eXchange) and OLIF (Open Lexicon Interchange Format), which originally was

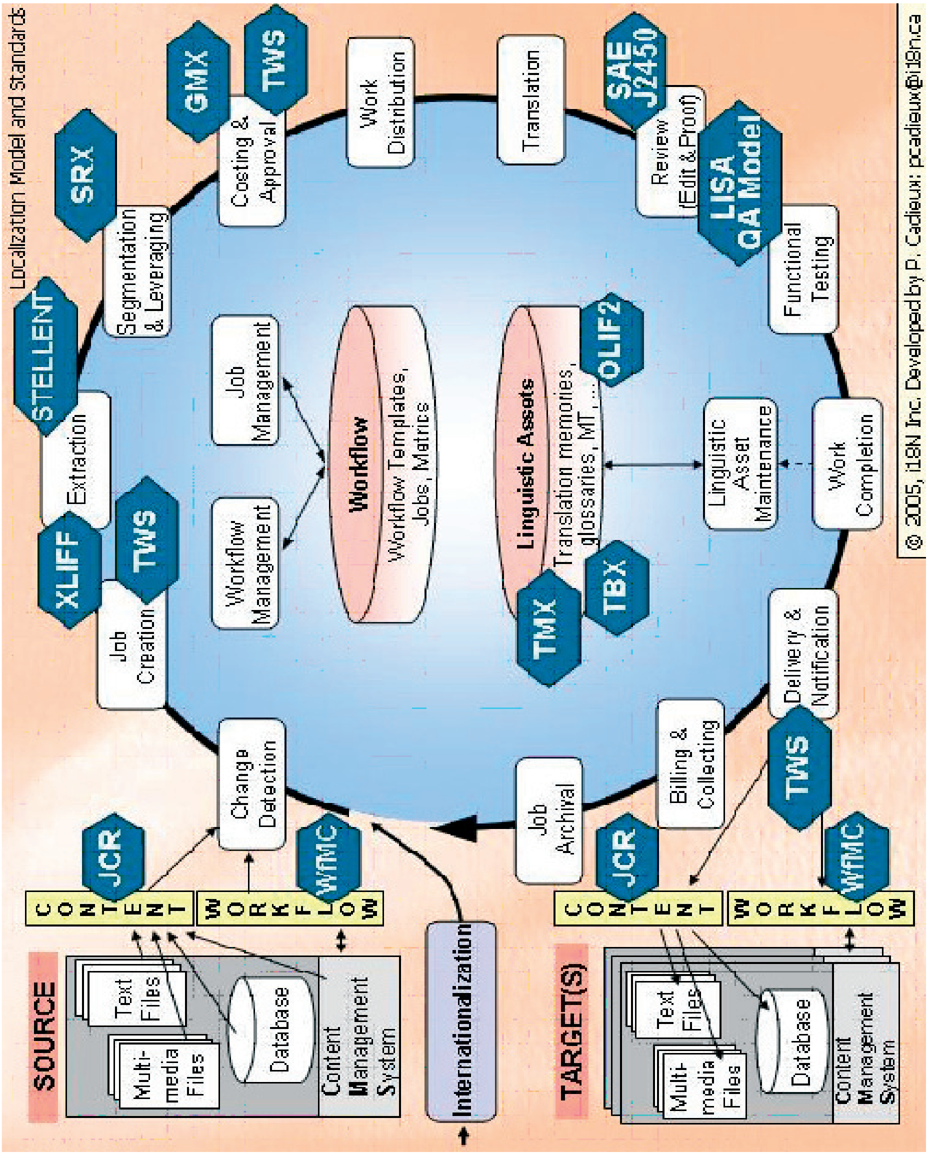


Fig. 1: Localization Model and Standards

mainly intended for the exchange of MT lexicons. In this context, he also mentions some of the difficulties which complicate the conversion of proprietary MT lexicons from and into OLIF. As the development of OLIF converters is a non-trivial task it is not surprising that MT vendors continue to use other exchange formats ranging from simple text files to complex proprietary formats.

Monica Gavrila, Walther von Hahn and Cristina Vertan present MANAGELEX, a generic lexicon management tool for creating, converting and merging lexicons which has been developed at Hamburg University. They outline the architecture of MANAGELEX and describe two of the modules which already have been implemented.

Georg Heeg discusses a software design approach to allow interchange of linguistic data. He focuses on the modelling of the linguistic concepts represented in the data and describes the transfer between exchange formats as a multi-tier interpretation/generation. The discussed concepts are implemented in Smalltalk, a programming environment enabling flexible conversion of data between formats supported by TMS.

Finally, Rachel Herwartz and Birgit Wöllbrink present a non-commercial internet discussion platform open to terminologists, translators and technical writers (*www.terminologieforum.de*) which was launched in January 2005.

The contributions in this issue show that the ultimate goal – blind interchange of terminological and lexicographical data – is still out of reach. Consequently, the development of suitable standards, which opens interesting perspectives for further research, is the objective of several ongoing research projects.

A list of important terminological and lexicological standards and research projects such as MILE can be found in the Appendix of this issue.

Endnote

[1] This model of the localization process was created by Pierre Cadieux, president of i18N Inc. (*www.i18n.ca*) and regular speaker at LISA events. The model has been used to describe and compare localization management systems and standards that apply to the localization process.

Uta Seewald-Heeg

Terminology Exchange without Loss?

Feasibilities and Limitations of Terminology Management Systems (TMS)

Abstract

The present article gives an overview over exchange formats supported by Terminology Management Systems (TMS) available on the market.

As translation is one of the eldest application domains for terminology work, most terminology tools analyzed here are components of computer-aided translation (CAT) tools.

In big corporates as well as in the localization industry, linguistic data, first of all terminology, have to be shared by different departments using different systems, a situation that can be best solved by standardized formats.

The evaluation of seven widely used TMS shows, however, that formats other than the standards proposed by organizations like LISA currently dominate the picture. In many cases, the only way to share data is to pass through flat structured data stored as tab-delimited text files.

1 Workflow and Interchange Scenarios

In the brief history of terminology management since the 1960s, when the first databases for terminology work were developed, terminology management has become a key resource, not only for the language industry, but also for globally acting industrial firms.

Usually, different departments within a company have access to the terminology resources, and if freelancers or translation service providers come into play, terminology interchange with external partners has to be organized as well.

At least in an architecture where corporate terminology has to be accessed from different applications under different circumstances – this is, for example, the case in corporates like SAP or DaimlerChrysler – questions of terminology interchange and supported formats arise. The need of interchange formats that guarantee the identification of data categories in different environments becomes obvious (Alder 1998). Here, standards come into play that map local system data categories to data categories specified in an open standard (Fig. 1), provided that developers of NLP tools make use of such standardized formats.

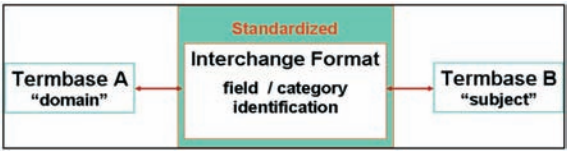


Fig. 1: Mapping local system categories to categories specified in a standard (following Alder 1998:12)

2 Interchanging Terminological Data – Standards

The need for terminology interchange has long been recognized by industrial users of TMS. Consequently, the past 15 years have seen several standardization initiatives aimed at developing standardized formats. One of these initiatives led to the CLS Framework (Melby/Wright 2000) which deals with the structure and content of terminological databases (Fig. 2). The CLS Framework (CLS stands for Concept-oriented with Links and Shared references, cf. Melby/Wright 1998) is based on the ISO 12620 standard "Computer applications in terminology – Data categories" which was published in 1999. CLS provides explicit data models for all types of terminological databases by structuring the items in a term entry according to theory and practice in concept-oriented terminology. The framework specifies the structure of a term entry and the relationships among data items in an entry using as one of the formats describing the structure of a terminological entry the Machine-Readable Terminology Interchange Format (MARTIF).

The development of the MARTIF standard, which formed the starting point for the CLS framework, was actually preceded by the development of OLIF (Open Lexicon Interchange Format), a more machine oriented standard, originally focussing on Machine Translation. The XML-compliant OLIF2 standard published in 2002 defines a large number of lexical features, but does not make statements about their structural embedding (Wittenburg/Gibbon/Peters 2001). Although OLIF2 aims at integrating data of Machine Translation and of Terminology Management Systems, OLIF has been of little importance in the field of Terminology Management Systems so far.

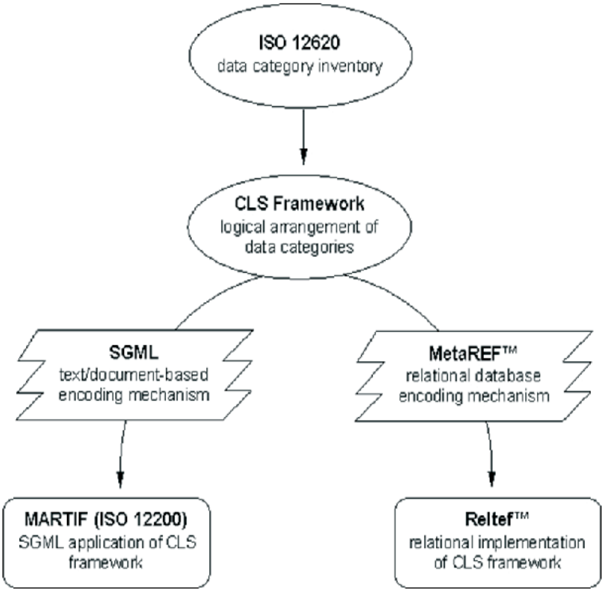

ISO 12620
data category inventory
CLS Framework
logical arrangement of data categories
SGML
text/document-based encoding mechanism
MetaREF™
relational database encoding mechanism
MARTIF (ISO 12200)
SGML application of CLS framework
Reltef™
relational implementation of CLS framework

Fig. 2: Structure of the CLS Framework (Melby/Wright 2000)

Another standard released to the public in 2002 by the Localization Industry Standards Association (LISA) is the TermBase eXchange Format (TBX) worked out by the LISA working group for the development and maintenance of open standards for the language industry, OSCAR (Open Standards for Container/Content Allowing Re-use). TBX, which is also based on XML, is only slowly being integrated into commercial terminology systems.

3 Terminology Management Systems (TMS)

3.1 Conceptual Features of TMS

Despite the existence of standards, commercial TMS still seem to be far away from the expressed goal of CLS, which is preservation of data when interchanging terminology (Alder 1998:6).

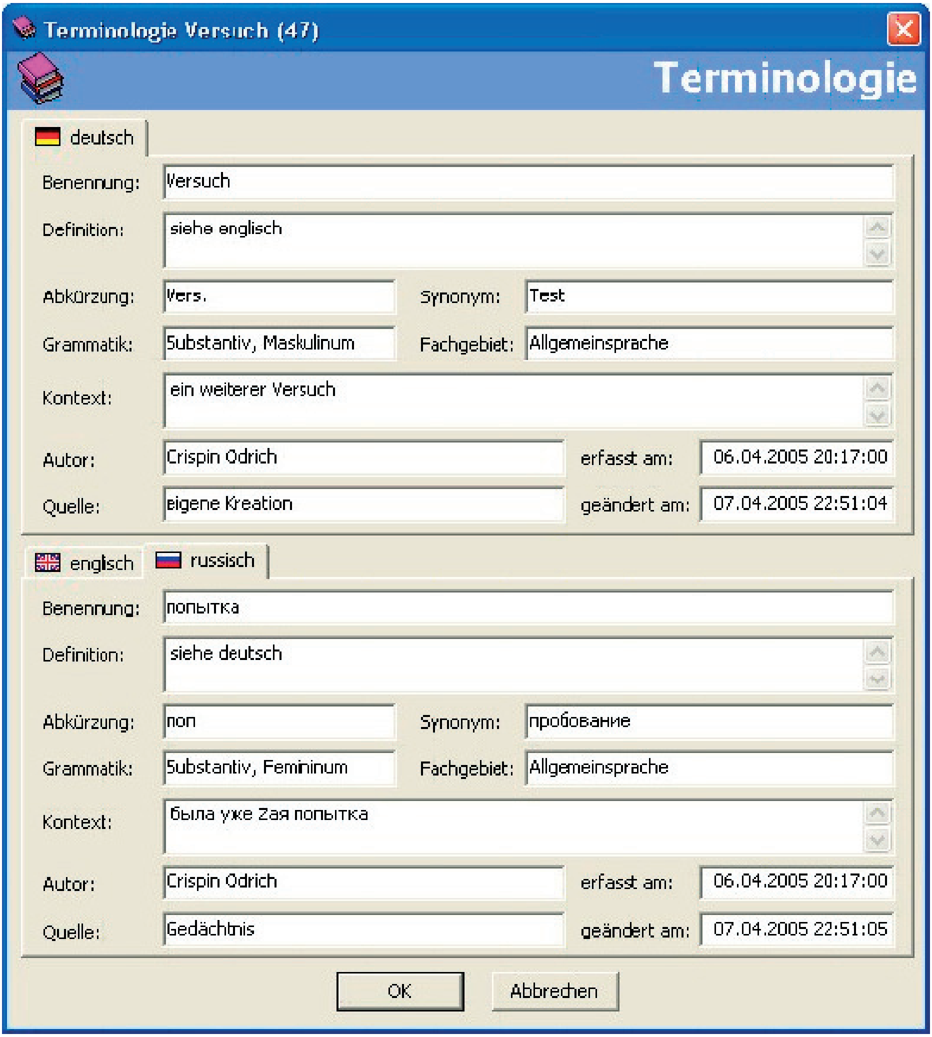
TMS not only differ in the formats they store lexical or terminological data, but also in their conceptual features. They can be classified by their

- **language concept** specifying whether a system is monolingual, bilingual, or allows multilingual data;
- **entry structure** which either can be predefined, definable or free, that is entirely specifiable by the user;
- **entry model** distinguishing systems only allowing a lemma-oriented structuring of the terminological database from systems allowing concept-oriented keeping of data;

Regarding the conceptual features of TMS the difference in the entry structure turns out to be one of the key problems.

3.2 Systems

In order to give an idea of the variety of differences concerning the conceptual features as well as the supported formats of existing commercial products, 7 systems have been selected. The following sections contain a discussion of their interchange functionalities according to the list below:



Benennung:	Versuch		
Definition:	siehe englisch		
Abkürzung:	Vers.	Synonym:	Test
Grammatik:	Substantiv, Maskulinum	Fachgebiet:	Allgemeinsprache
Kontext:	ein weiterer Versuch		
Autor:	Crispin Odrich	erfasst am:	06.04.2005 20:17:00
Quelle:	eigene Kreation	geändert am:	07.04.2005 22:51:04

Benennung:	попытка		
Definition:	siehe deutsch		
Abkürzung:	поп	Synonym:	пробование
Grammatik:	Substantiv, Femininum	Fachgebiet:	Allgemeinsprache
Kontext:	была уже 2ая попытка		
Autor:	Crispin Odrich	erfasst am:	06.04.2005 20:17:00
Quelle:	Gedächtnis	geändert am:	07.04.2005 22:51:05

Fig. 3: GFT DataTerm interface

GFT DataTerm by GFT (*www.gft-online.de*).
UniTerm by Acolada (*www.acolada.de*).
Déjà Vu Terminology by Atril (*www.atril.com*).
SDL TermBase by SDL (*www.sdl.com*).
MultiTerm iX by SDL Trados (*www.trados.com*).
TermStar XV by Star (*www.star-group.net*).
crossTerm by across (*www.across.net*).

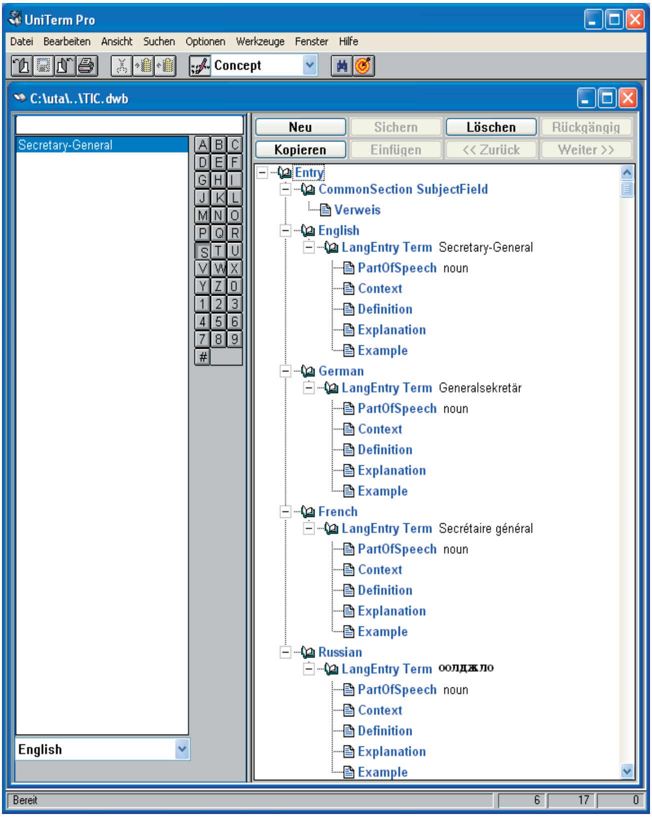
3.2.1 Standalone Systems

The first system mentioned here, **GFT DataTerm** (Fig. 3), is a standalone system in the sense that it does not provide interfaces to tools like Translation Memories (TM) or other applications. It is a lemma-oriented system, even if multiple language pairs can be stored in a single entry. Descriptive categories can only be assigned to individual terms; other levels of specification, e.g. a concept level linking different terms to a given concept do not exist. For import, GFT DataTerm provides tab-delimited text file format as well as the Excel XML spreadsheet format. Formats provided for the export of terminology are Excel and XML-based MARTIF.

Another standalone system is the **UniTerm** tool (Fig. 4) from which terminological data can also be exported as text file or as XML together with a DTD[1]. It has a definable entry structure and allows multilingual conceptual information. Term describing fields as well as fields containing conceptual information can be selected among a predefined set of categories which can be labelled individually. Furthermore, for different purposes of terminological work different editing patterns are available.

3.2.2 Integrated Systems

In contrast to the standalone systems mentioned so far, most terminology systems are actually integrated into TM environments. Thus, across, Déjà Vu, SDLX, Star, and Trados all have more or less powerful terminology components. In

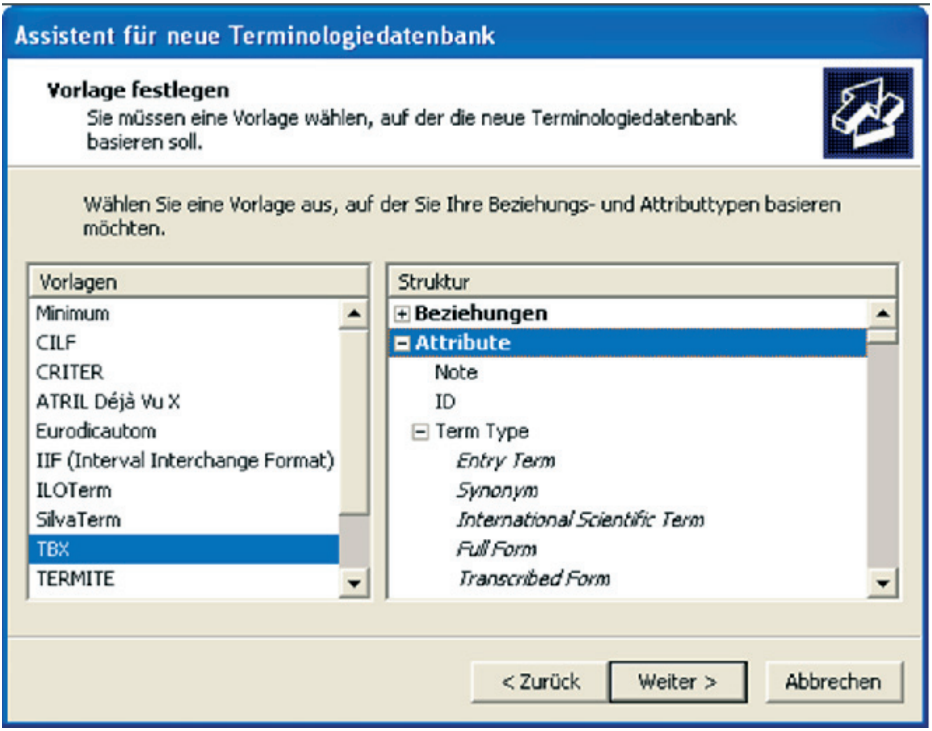


Secretary-General

English

Entry
CommonSection SubjectField
Verweis
English
LangEntry Term Secretary-General
PartOfSpeech noun
Context
Definition
Explanation
Example
German
LangEntry Term Generalsekretär
PartOfSpeech noun
Context
Definition
Explanation
Example
French
LangEntry Term Secrétaire général
PartOfSpeech noun
Context
Definition
Explanation
Example
Russian
LangEntry Term оолджло
PartOfSpeech noun
Context
Definition
Explanation
Example

Fig. 4: UniTerm interface



Assistent für neue Terminologiedatenbank

Vorlage festlegen

Sie müssen eine Vorlage wählen, auf der die neue Terminologiedatenbank basieren soll.

Wählen Sie eine Vorlage aus, auf der Sie Ihre Beziehungs- und Attributtypen basieren möchten.

Fig. 5: Pattern selection for the structure of entries in Déjà Vu

the case of the Star and the Trados products, i.e. TermStar and MultiTerm, the terminology components can even be purchased separately.

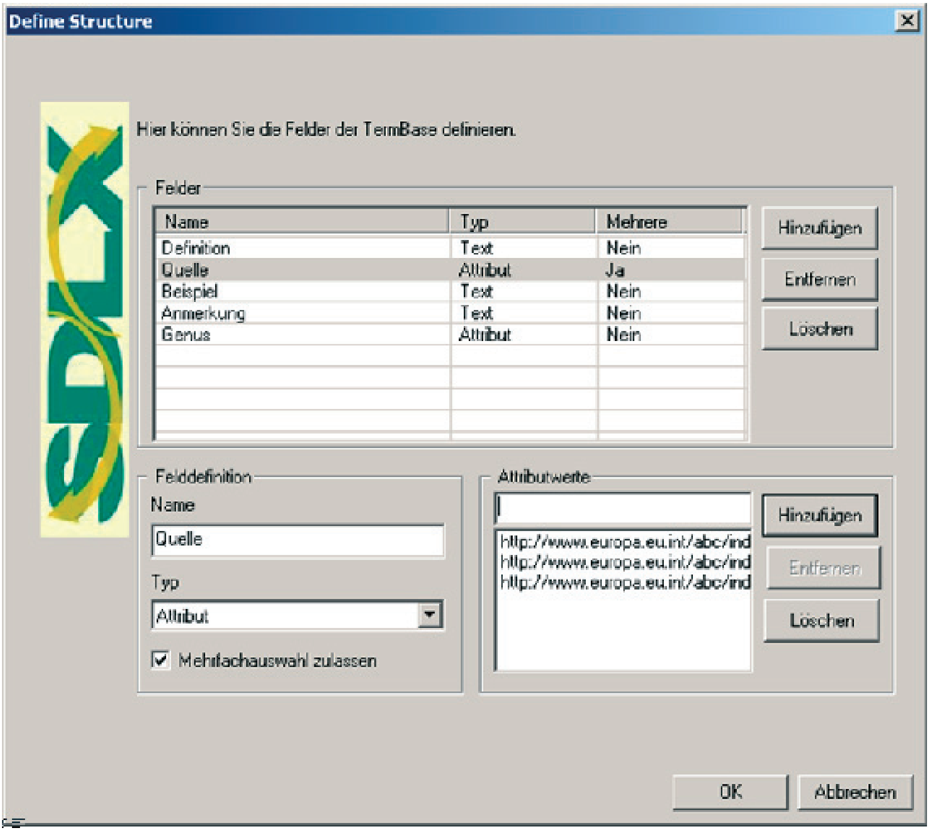
Part of the **Déjà Vu** TM-System is a so-called terminology database which is mainly lemma-oriented. To create a termbase, Déjà Vu provides templates to determine the entry structure for a new database. One of them reflects the structure and categories of TBX (Fig. 5) although TBX is not supported for import or export. Déjà Vu allows the import of text files, Excel and Access files as well as TermStar files. The same file types can also be exported.

When terminology has to be imported from an Excel file, the Excel column headers have to be assigned to Déjà Vu fields, a common way to map the content of the spreadsheet file to the terminology system where the user has to determine the fields to be imported and to specify whether filters shall be applied.

The **SDL TermBase** (Fig. 6), a component of the SDLX TM system, is structured very similarly to the Déjà Vu terminology component. As far as the multilinguality and the treatment of synonyms are concerned, the structuring of the data is concept-oriented. But one misses a conceptual level allowing the specification of non-redundant information valid for the concept, that is, for all terms of a given entry. For the import and export of terminology, apart from the proprietary format, tab-delimited text files as well as files in Trados MultiTerm 5 format can be imported.

The Trados terminology component **MultiTerm iX** is one of the two terminology systems which provide interfaces to other components of a translation memory environment, but which can also be used without launching the TM system.

MultiTerm provides a concept-oriented storage of data (Fig. 7) and has a hierarchical structure with three different levels, one level to specify concept-related information, another one for language-specific terminological information, and a third one to describe an individual term. It has a definable entry structure, but provides also predefined termbase templates in which the fields are already specified, and the entry structure is already defined. The structure of the termbank and the terminological data are stored in separate files. For import, MultiTerm supports Excel and tab-delimited text files which first have to be converted by MultiTerm Convert (Fig. 8). For export, MultiTerm provides as format its own XML format which follows the main structuring principles of TBX although it proved to be incompatible with TBX in the evaluated version (Trados 7). Apart from its own XML format, MultiTerm IX provides two other formats for terminology export, MultiTerm 5 and tab-delimited text file format.



Define Structure

Hier können Sie die Felder der TermBase definieren.

Felder

Name	Typ	Mehrere
Definition	Text	Nein
Quelle	Attribut	Ja
Beispiel	Text	Nein
Anmerkung	Text	Nein
Genus	Attribut	Nein

Felddefinition

Name: Quelle

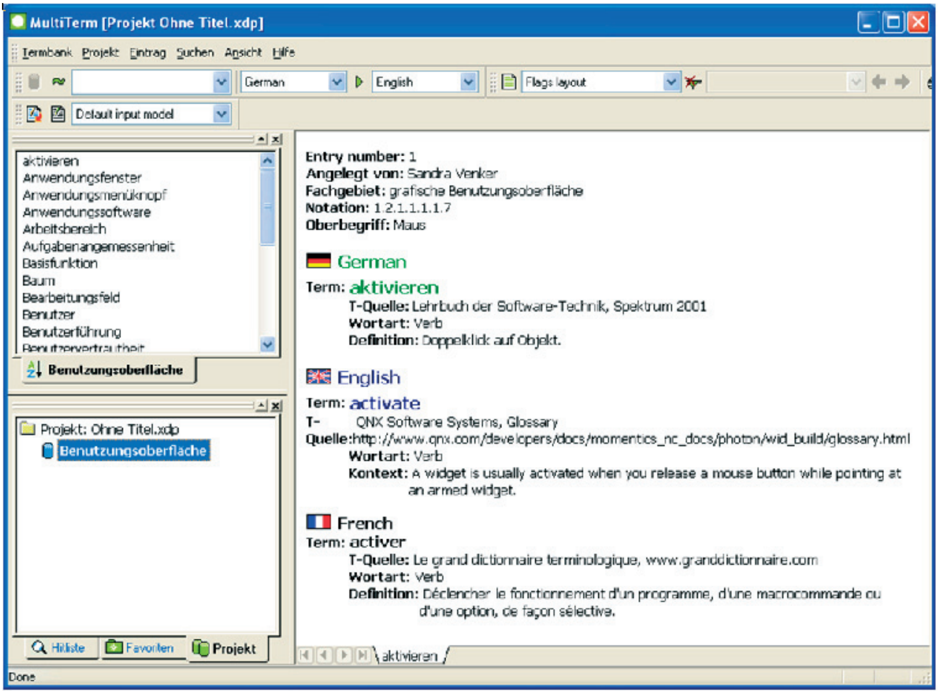
Typ: Attribut

☑ Mehrfachauswahl zulassen

Attributwerte

http://www.europa.eu.int/abc/ind
http://www.europa.eu.int/abc/ind
http://www.europa.eu.int/abc/ind

Fig. 6: Definition of termbank structure in SDLX



Entry number: 1
Angelegt von: Sandra Venker
Fachgebiet: grafische Benutzungsoberfläche
Notation: 1.2.1.1.1.1.7
Oberbegriff: Maus

German

Term: aktivieren
T-Quelle: Lehrbuch der Software-Technik, Spektrum 2001
Wortart: Verb
Definition: Doppelklick auf Objekt.

English

Term: activate
T-Quelle: QNX Software Systems, Glossary http://www.qnx.com/developers/docs/momentics_nc_docs/photon/wid_build/glossary.html
Wortart: Verb
Kontext: A widget is usually activated when you release a mouse button while pointing at an armed widget.

French

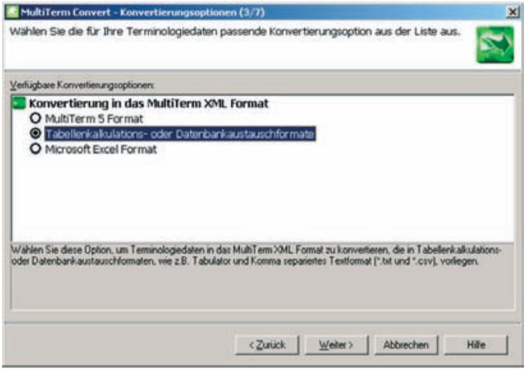
Term: activer
T-Quelle: Le grand dictionnaire terminologique, www.granddictionnaire.com
Wortart: Verb
Definition: Déclencher le fonctionnement d'un programme, d'une macrocommande ou d'une option, de façon sélective.

Fig. 7: MultiTerm iX interface

The Star terminology system, **TermStar XV**, is the other TMS which provides interfaces to other components of a translation environment, and which can also be used as a standalone system, i.e. independent of a translation memory environment. TermStar has a definable entry structure, however with a predefined set of possible data categories which can be named according to the need of the users.

Similar to MultiTerm, TermStar (see Fig. 9) distinguishes different description levels: The header of an entry is meant to store conceptual information. Terms can be described depending on the individual language, and an intermediate information level can be used to store information for all terms of a given language.

For the import of terms TermStar provides, apart from its proprietary formats of different TermStar versions, an XML-based MARTIF and for everything else an import dialogue for so



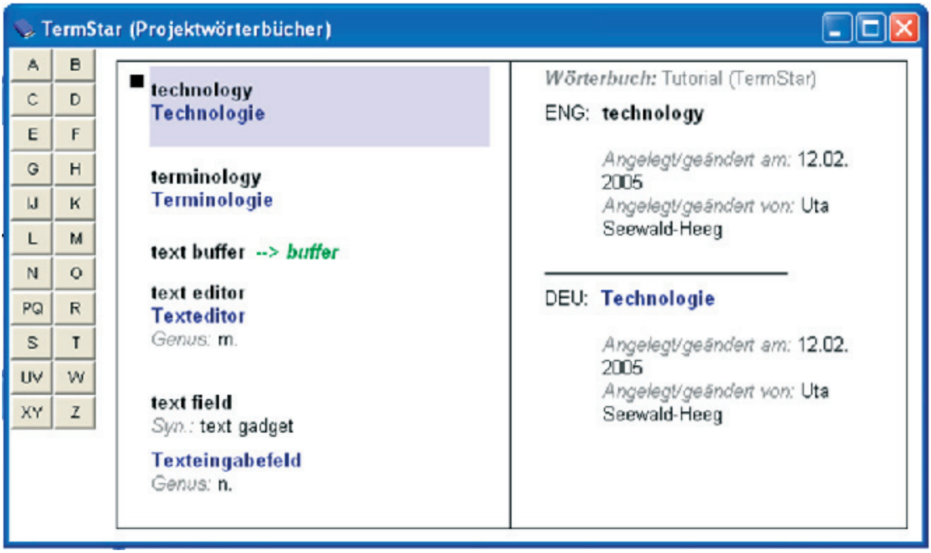
Wählen Sie die für Ihre Terminologiedaten passende Konvertierungsoption aus der Liste aus.

Verfügbare Konvertierungsoptionen:

Konvertierung in das MultiTerm XML Format
○ MultiTerm 5 Format
● Tabellenkalkulations- oder Datenbankaustauschformate
○ Microsoft Excel Format

Wählen Sie diese Option, um Terminologiedaten in das MultiTerm XML Format zu konvertieren, die in Tabellenkalkulations- oder Datenbankaustauschformaten, wie z.B. Tabulator und Komma separiertes Textformat (*.txt und *.csv), vorliegen.

Fig. 8: Format conversion using MultiTerm Convert


TermStar (Projektwörterbücher)
A B C D E F G H IJ K L M N O PQ R S T UV W XY Z

■ technology
Technologie

terminology
Terminologie

text buffer --> *buffer*

text editor
Texteditor
Genus: m.

text field
Syn.: text gadget

Texteingabefeld
Genus: n.

Wörterbuch: Tutorial (TermStar)

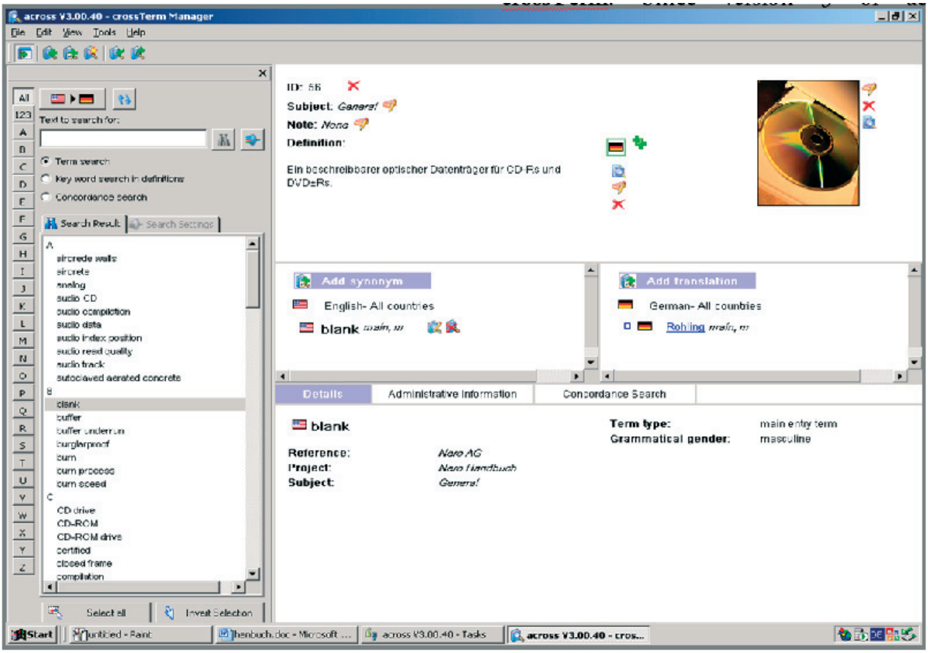
ENG: technology

Angelegt/geändert am: 12.02. 2005
Angelegt/geändert von: Uta Seewald-Heeg

DEU: Technologie

Angelegt/geändert am: 12.02. 2005
Angelegt/geändert von: Uta Seewald-Heeg

Fig. 9: TermStar XV interface


across V3.00.40 - crossTerm Manager
File Edit View Tools Help
All 123 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Search Result
Search Settings
Select all
Invert Selection
Add synonym
Add translation
Details
Administrative information
Concordance Search
Start
untitled - Paint
handbuch.doc - Microsoft ...
across V3.00.40 - Tasks
across V3.00.40 - cros...

Text to search for:

● Term search
○ Key word search in definitions
○ Concordance search

A
aircrete walls
aircrete
analog
audio CD
audio compilation
audio data
audio index position
audio read quality
audio track
autoclaved aerated concrete
B
blank
buffer
buffer underrun
burglarproof
burn
burn process
burn speed
C
CD drive
CD-ROM
CD-ROM drive
certified
closed frame
compilation

ID: 56
Subject: *General*
Note: *None*
Definition:

Ein beschreibbarer optischer Datenträger für CD-Rs und DVD±Rs.

English- All countries
blank *main, m*

German- All countries
Rohling *main, m*

blank

Reference: *Nero AG*
Project: *Nero Handbuch*
Subject: *General*

Term type: main entry term
Grammatical gender: masculine

Fig. 10: crossTerm user interface

called “user defined formats”, which allows, for example, to configure the import of Excel and MultiTerm 5 files. If proprietary formats are not considered, the export from TermStar is restricted to XML MARTIF.

Among the systems mentioned here, the most recent system on the market is across, a translation management environment which also provides a terminology component called **crossTerm**. Since version 3 of across, crossTerm allows concept-oriented data storage. Concept-relevant information can be stored in the head of an entry which is visually separated from the bilingual view of an entry (Fig. 10). The across developers have avoided using a proprietary terminology format. In crossTerm, terminology is stored in TBX format, which is also the only format provided for export. To import data crossTerm provides in addition to CSV-format, the Langenscheidt electronic dictionary format, Trados MultiTerm 5, and the Star MARTIF format.

4 Supported formats

The evaluation has shown that all the systems analyzed so far allow import from Excel files or file formats such as CSV or TXT that can be generated by Excel. As Trados – at least until its acquisition by SDL – has dominated the TM and TMS market, several products also support MultiTerm format. However, instead of supporting MultiTerm iX, they usually support the text based format formerly used by Trados 5. The support of formats can be visualized as illustrated below (see Fig. 11).

5 Exchange of data

As shown in Figure 11, Excel or Excel-derived formats like CSV and tab-delimited text are in many cases the only formats allowing the interchange of data between two or more systems. Thus, the question arises whether all of the data intended to be transferred are actually transferred or interchanged completely and correctly using Excel

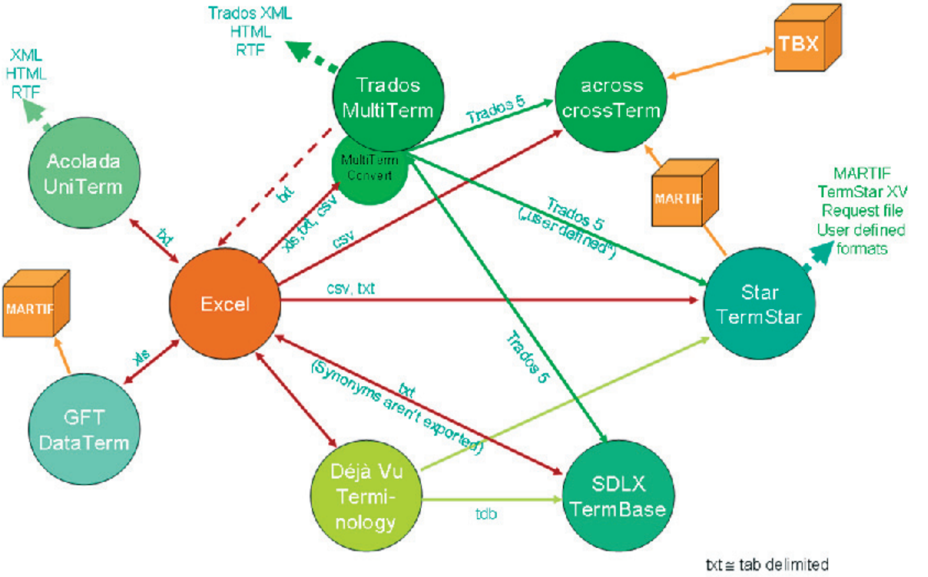
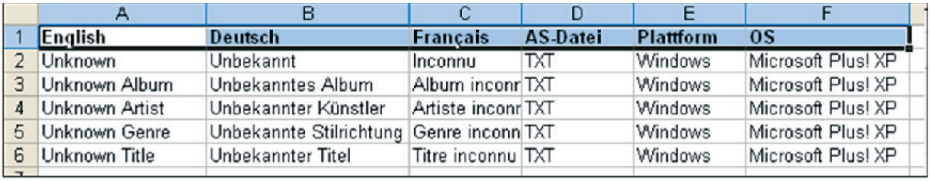


Fig. 11: Exchange formats supported by TMS



	A	B	C	D	E	F
1	English	Deutsch	Français	AS-Datei	Plattform	OS
2	Unknown	Unbekannt	Inconnu	TXT	Windows	Microsoft Plus! XP
3	Unknown Album	Unbekanntes Album	Album inconr	TXT	Windows	Microsoft Plus! XP
4	Unknown Artist	Unbekannter Künstler	Artiste inconr	TXT	Windows	Microsoft Plus! XP
5	Unknown Genre	Unbekannte Stilrichtung	Genre inconn	TXT	Windows	Microsoft Plus! XP
6	Unknown Title	Unbekannter Titel	Titre inconnu	TXT	Windows	Microsoft Plus! XP

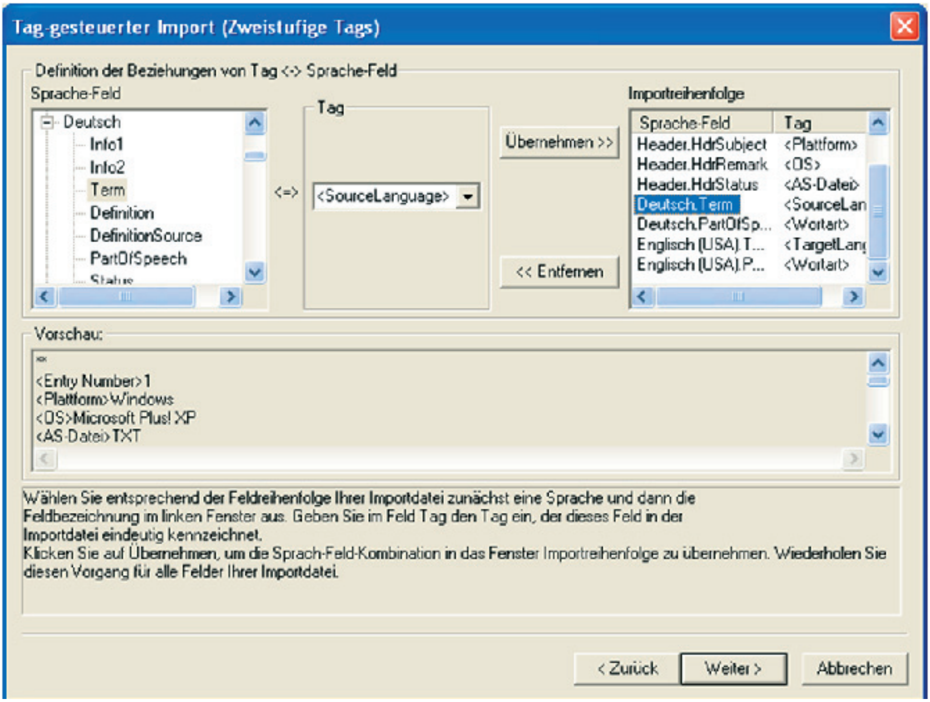
Fig. 12: Multilingual glossary in Excel format

files. To answer this question the structural layer comes into play, because each system presupposes a defined structuring of the stored data. And as data interchange also has to guarantee the correct interpretation of the content, we also have to consider the semantic, or representational layer.

To gain insight in this question, we now will have a closer look at the import and export of terminology stored in Excel files as well as the interchange of these data between different TMS.

The starting point will be an Excel file containing a simple multilingual glossary (Fig. 12) in the form glossaries are provided by Microsoft with some additional information.

In order to get these data into **MultiTerm iX**, they first have to be converted by MultiTerm Convert into MultiTerm-compatible format. During this process, the Excel column headers have to be assigned to MultiTerm fields, and the entry structure has to be defined. The result of



Sprache-Feld	Tag
Header.HdrSubject	<Plattform>
Header.HdrRemark	<OS>
Header.HdrStatus	<AS-Datei>
Deutsch.Term	<SourceLan
Deutsch.PartOfSp...	<Wortart>
Englisch (USA).T...	<TargetLan(
Englisch (USA).P...	<Wortart>

```
**
<Entry Number>1
<Plattform>Windows
<OS>Microsoft Plus! XP
<AS-Datei>TXT
```

Wählen Sie entsprechend der Feldreihenfolge Ihrer Importdatei zunächst eine Sprache und dann die Feldbezeichnung im linken Fenster aus. Geben Sie im Feld Tag den Tag ein, der dieses Feld in der Importdatei eindeutig kennzeichnet.
Klicken Sie auf Übernehmen, um die Sprach-Feld-Kombination in das Fenster Importreihenfolge zu übernehmen. Wiederholen Sie diesen Vorgang für alle Felder Ihrer Importdatei.

Fig. 13: Import dialogue in TermStar

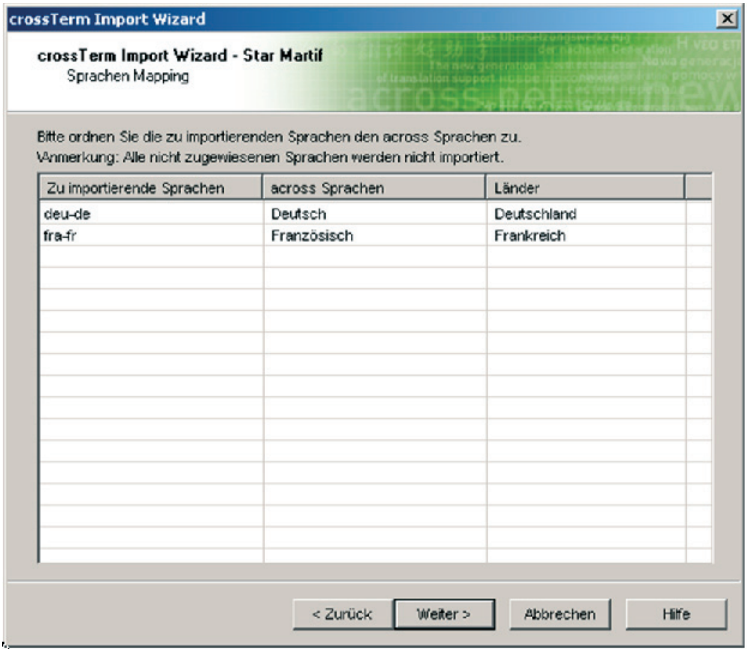
the conversion is a termbank definition file, and an XML file containing the terminological data. This XML file finally can be used to create a new termbase and to launch the default process for the import. Importing data from an Excel file produces a satisfactory result, since all information can be transferred completely and correctly.

Furthermore, some of the term-related information may not be present in all of the entries. In this case, the use of MultiTerm is problematic, because the MultiTerm export functionality creates files where descriptive fields, which are used only in part of the entries, are ignored when writing the tab-delimited text file. As a result, the system generates columns with different type of content in their respective cells. In this case, the resulting files turn out to be unusable for further handling. Another kind of problem is caused by line breaks in definition texts. As line breaks split up an entry on different lines, an import where one line corresponds to one entry is not possible any more.

The export of data in tab-delimited format does not necessarily suffer from these limita-tions. If unused fields of the entry structure are exported as empty fields (this is, for exam-ple, the case when exporting data in tab-delimited text format from the **SDL TermBase**) the structure of the content can be preserved, so that the export file allows further handling of the exported data and import in other systems supporting tab-delimited format.

Another export scenario is the exchange of terminology between **MultiTerm iX** users and users of other systems supporting MultiTerm 5 format including terminologists still working with Trados 5. A closer look at the MultiTerm 5 export functionality provided by MultiTerm iX revealed that this functionality supports only bilingual export. As a result, a multilingual termbase can only be exported selecting different language pairs with one language as reference. Therefor, n languages require n-1 export procedures, and certainly also n-1 import procedures on the side of the receiving system.

Saved as an ANSI-encoded tab-delimited text file, an Excel glossary can also be imported in



Bitte ordnen Sie die zu importierenden Sprachen den across Sprachen zu.
Anmerkung: Alle nicht zugewiesenen Sprachen werden nicht importiert.

Zu importierende Sprachen	across Sprachen	Länder
deu-de	Deutsch	Deutschland
fra-fr	Französisch	Frankreich

Fig. 14: crossTerm Import Wizard for Star MARTIF

TermStar XV, where the column headers have to be assigned to TermStar fields (Fig. 13). No information is lost during this process; the only inconvenience is that the column headers of the text file are imported in TermStar as first entry.

The import of MultiTerm 5 files to TermStar XV has to pass through the conversion of the MultiTerm 5 text file in ANSI format because – at least in the build analyzed here – Unicode-encoded MultiTerm files are not supported which already restricts the type of languages which can be interchanged with this format. The MultiTerm 5 import in TermStar transfers the entire information to TermStar.

The import of the Excel file in **crossTerm** leads to a satisfactory result as it did for the previously mentioned systems.

The import of a Star MARTIF file into crossTerm does not differ substantially from the Excel import, i.e. the field names of both representations have to be mapped to each other (Fig. 14). Here again, the result is quite satisfactory.

From a purely technical point of view, terminological data can be imported, exported, and interchanged using tab-delimited text files. However, as systems like MultiTerm allow a certain descriptive field to be used at different levels and related to distinct fields, the information of the embedding of categories disappears when mapping entry structures to flat rows and columns so that this kind of information cannot be maintained transferring data between different systems using tab-delimited text format.

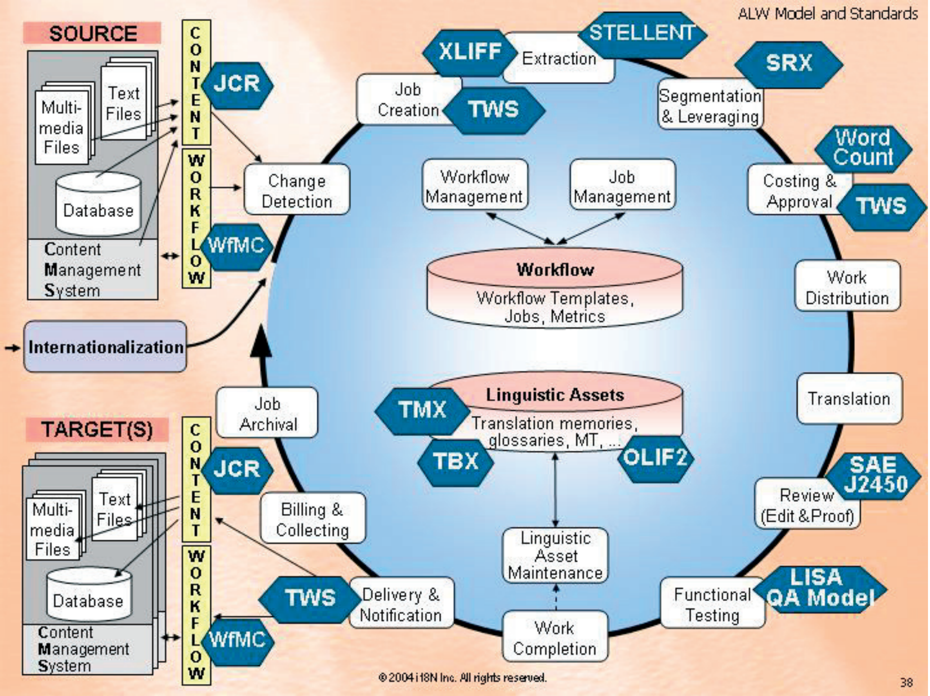


Fig. 15: The role of standards in an automated workflow

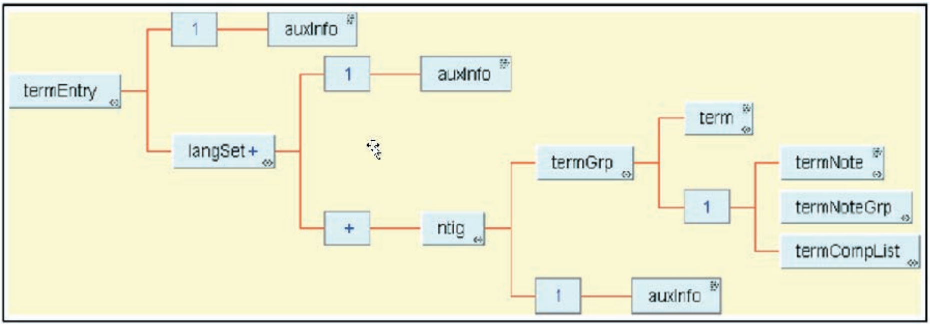


Fig. 16: Structure of a terminological entry in TBX

6 The Role of Standards in an Automated Workflow

The interchange scenario described in the previous sections calls for standardized interchange between NLP systems. There are already workflow scenarios where the only way of cost-effective and efficient transfer of data from one tool to another and from one phase to another consists of using standardized formats. This is, for example, the case in software localization where standards play a predominant role in the localization process (see Fig. 15)[2].

Concerning terminology interchange the Localization Industry Standards Association (LISA) propagates TBX. TBX is an XML-based terminology markup format that is consistent with ISO 12200 (MARTIF).

A TBX file consists of a header that describes the file, a set of entries, one per concept in the termbase, and a set of terms for each concept, which designate the concept, and which are grouped by language. Thus, the structure of a terminological entry in the body of a TBX document distinguishes three levels (see Fig. 16): the entry level (<termEntry>), the language level (<LangSet>), and the term level (<ntig>). TBX therefore provides all prerequisites for supporting concept-oriented terminology work and guarantees a number of benefits for terminology exchange provided that it is supported by more than one commercial system.

7 Conclusion

We have to conclude that standardized interchange formats for platform-independent terminology interchange are still rarely supported by commercial systems. Regarding the supported import formats of terminology systems, CSV instead of TBX turns out to be a quasi-standard at least if we use the number of systems supporting this format as an indicator. The export to CSV or tab-delimited files may, however, be problematic when line breaks occur in descriptive text fields, or when the number of descriptive fields used differs between several entries, as could be seen in the case of the MultiTerm iX export. Here, reusable data are only generated if the type and number of information describing an entry is homogeneous over all entries. Another problem may occur if the structuring and the number of fields used in the entry structure of one system is not compatible with the number of fields allowed in the receiving system.

There is no doubt that standards are indispensable, not only from the point of view of the user, but also with respect to complex workflow scenarios. Perhaps, new industrial alliances as they were formed in 2005 will enforce the support of open source formats. From the point of view of the terminologist as well as from the point of view of the company which has to handle terminology in complex workflow situations the limited use of standards in terminology exchange by commercial systems is rather disillusioning.

References

ALDER, A. C. (1998). "An Experiment in Blind Terminology Interchange: Developing and Testing Conversion Algorithms for Externally Supplied Data". Master Thesis, Brigham Young University.

LISA (Localization Industry Standards Association). *http://www.lisa.org* [13.01.2006].

MELBY, A./ WRIGHT, S. E. (1998). "The CLS Framework Overview". *http://www.ttt.org/clsframe/overview.html* [19.01.2006].

MELBY, A. / WRIGHT, S. E. (2000). "The CLS Framework". *http://www.ttt.org/clsframe/index.html* [9.11.2005].

OLIF (Open Lexicon Interchange Format).*http://www.olif.net/* [19.01.2006]

OSCAR (Open Standards for Container/Content Allowing Re-use). *http://www.lisa.org/sigs/oscar/* [13.01.06].

TBX (TermBase eXchange). http://www.lisa.org/standards/tbx/ [13.01.2006].

WITTENBURG, P. / GIBBON, D. / PETERS, W. (2001): "Metadata Elements for Lexicon Descriptions". IMDI1 Technical Report. *http://www.mpi.nl/ISLE/documents/draft/ISLE_Lexicon_1.0.pdf* [16.01.2006]

ZENK, W. (2006): "UniTerm – Formats and Terminology Exchange". In: Geldbach, St., Seewald-Heeg U. (eds.): "Exchange of Lexical and Terminological Resources", LDV-Forum 21(1), pp 19-26.

TMS Vendors

Acolada (*www.acolada.de*) [13.01.2006].

across (*www.across.net*) [13.01.2006].

Atril (*www.atril.com*) [13.01.2006].

GFT (*www.gft-online.de*) [13.01.2006].

SDL (*www.sdl.com*) [13.01.2006].

Star (*www.star-group.net*) [13.01.2006].

Trados (*www.trados.com*) [13.01.2006].

Endnotes

[1] For a detailed discussion of UniTerm, see also the contribution by ZENK in this volume.

[2] This model of the localization process was created by PIERRE CADIEUX, president of i18N Inc. (*www.i18n.ca*).

Wolfgang Zenk

UniTerm – Formats and Terminology Exchange

Abstract

This article presents UniTerm, a typical representative of terminology management systems (TMS). The first part will highlight common characteristics of TMS and give further insight into the UniTerm entry format and database design.

Practise has shown that automatic, i.e. blind exchange of terminologies is difficult to achieve. The second section gives criteria where the exchange between different TMS can fail and points out the relationship between the UniTerm like TMS data formats and existing terminology standards.

Finally, it will be discussed what requirements have to be met in order to enable a deeper integration of terminology standards in a TMS and thus also a smoother transition between different TMS. These requirements are evaluated with Acolada´s next generation TMS UniTerm Enterprise.

1 UniTerm Development

The UniTerm TMS has been inspired by two preceding product developments. These two products – Dictionary Workbench and Linguistic Resource Database (LRD) Editor – equally provide the source code basis upon which UniTerm has been built. These two applications can be characterized as follows:

Dictionary Workbench: a lexicographic tool for dictionary management and production. Dictionary Workbench has been used for specialist dictionaries from 1994 onwards.

LRD Editor: a TMS designed and developed within the scope of the EURAMIS project[1]. The LRD Editor has been developed between 1994 and 1998.

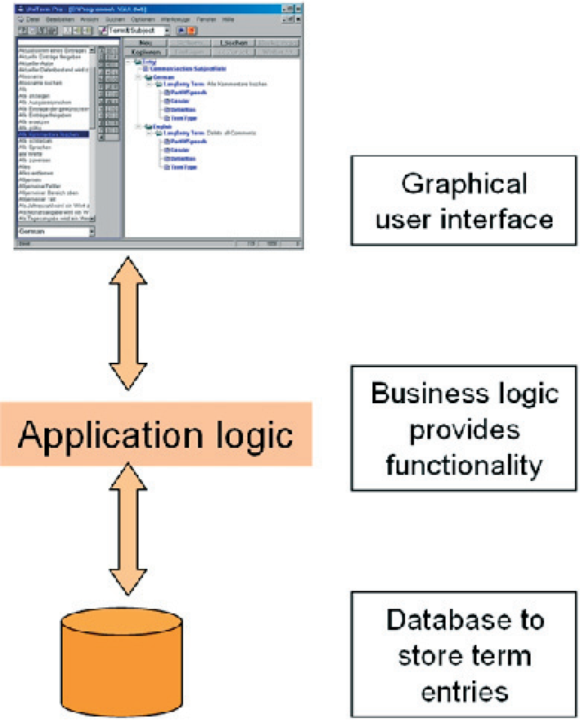

Graphical user interface
Application logic
Business logic provides functionality
Database to store term entries

Fig. 1: Typical software architecture for terminology management systems as database applications

Since 1999, the source code of these two systems has been unified to create the UniTerm system. Today, UniTerm offers the functionality of a full-fledged TMS. With a flexible implementation of different entry formats and additional tools for the dictionary production, the current UniTerm Pro version is used for terminological as well as for lexicographical work.

2 Characterization / System Architecture

UniTerm is essentially a database tool. This general architecture of the UniTerm TMS can be applied to almost all TMS. On top of the database layer are two further layers for application logic and a graphical user interface so that the software architecture can be characterized as a 3-tier model (see Figure 1).

In a 3-tier model database, application logic and user interface are implemented in different layers. By enabling a communication between each of these layers, changes in one layer may be made without causing implications to other layers and the whole software functionality.

The UniTerm system architecture can be applied to almost any TMS:

Database approach: searching with different search criteria and sorting of entries in different languages are crucial operations in terminological data which can be best performed by a database.

At the **user interface**, templates are offered to enter data. Preview functions provide a more user-friendly and less technical view on the data. The possibility to adapt templates and the structure of entries are directly linked to the database model implemented.

If this kind of system architecture is applicable to all standard TMS, what are distinctive criteria between different TMS? TMS usually differ in following features:

Range of languages: TMS support different numbers of languages. The treatment of languages with different coding and the support of Unicode are the most relevant questions.

Flexibility of the entry structure: The more advanced a TMS is, the less rigid the entry structure and the more adaptable editing templates become.

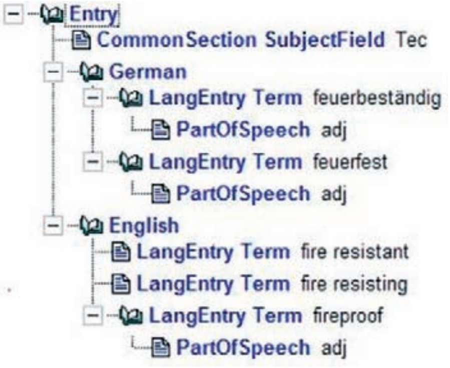
Database operations such as simple headword search, full-text search, searching in the structure (e.g. all nouns), filter functions and other special search functionalities (e.g. in UniTerm, it is possible to search for all entries that do not have a translation in a specified language).

3 UniTerm Entry Structure Design

3.1 UniTerm Entry Structure

With regard to entry formats, TMS are generally categorized into TMS with fixed formats (the format is predefined by the TMS vendor) and TMS with definable or free formats which need to be defined by the users themselves.

UniTerm is closer to a TMS with fixed format even though a number of data fields is offered to extend the entry structure with user-defined data fields. Experienced users may even implement their custom format.



```
Entry
  CommonSection SubjectField Tec
  German
    LangEntry Term feuerbeständig
      PartOfSpeech adj
    LangEntry Term feuerfest
      PartOfSpeech adj
  English
    LangEntry Term fire resistant
    LangEntry Term fire resisting
    LangEntry Term fireproof
      PartOfSpeech adj
```

Fig. 2: Sample entry coding in UniTerm

The UniTerm format follows a concept model with a common section in which the entry concept is described and the language section in which a term of a language is described. UniTerm allows the language section to be repeated any number of times to allow any number of languages to be used in a multilingual database but also to allow more than one term of a language to be added. As a consequence, the full entry structure can be used to describe each term.

The entry structure in UniTerm is illustrated in Figure 2 and can be characterized as follows:

1. The structure tries to provide a **superset of permissible data categories**. This approach follows the idea underlying terminology standards, i.e. users are allowed to select a suitable subset and create their own editing template. Such an editing template can be extended with further data categories and languages at any time without having to amend the database or a database definition.
2. The entry structure is built on **data categories provided by ISO 12620**[2].
3. **Additional data fields** have been introduced for translation memory and controlled language integration.
4. Finally, so-called **user fields** have been introduced. These data fields add further flexibility if a data category is to be defined which is not included in the default format.

Since its first version, the UniTerm entry format has been revised and extended in subsequent program versions. The current entry structure provides following features:

Increased flexibility: the values of data fields which were previously provided in a pre-defined list of values can be edited. Take, for example, the data field *Normative Authorization*. Its values (standardized term, preferred term, admitted term, deprecated term, superseded term, legal term, regulated term) had formerly been pre-defined as fixed values and can now be altered or edited by the user. This feature increases the flexibility on the one hand but has negative impact on (blind) terminology exchange on the other hand.

Increased usability: hierarchical levels that had been introduced below term level (e.g. grammar, term classification, concept related description) have been deleted. Entry templates thus become more readable and easier to work with.

Adaptations to new software development: data fields in a TMS are always of a certain type, e.g. provide a list of values from which a user selects one or more, provide system fields that hold administrative information, etc. In a similar way, data field types provided by UniTerm have following properties:

a) **system fields** which are automatically filled in by the system (e.g. creation date, update author, etc.);
b) **files** which allow to insert a reference to an external graphic, an audio file or a text file (RTF);
c) **text fields** where the user adds text;
d) **list values** usually are pre-defined and user-editable.

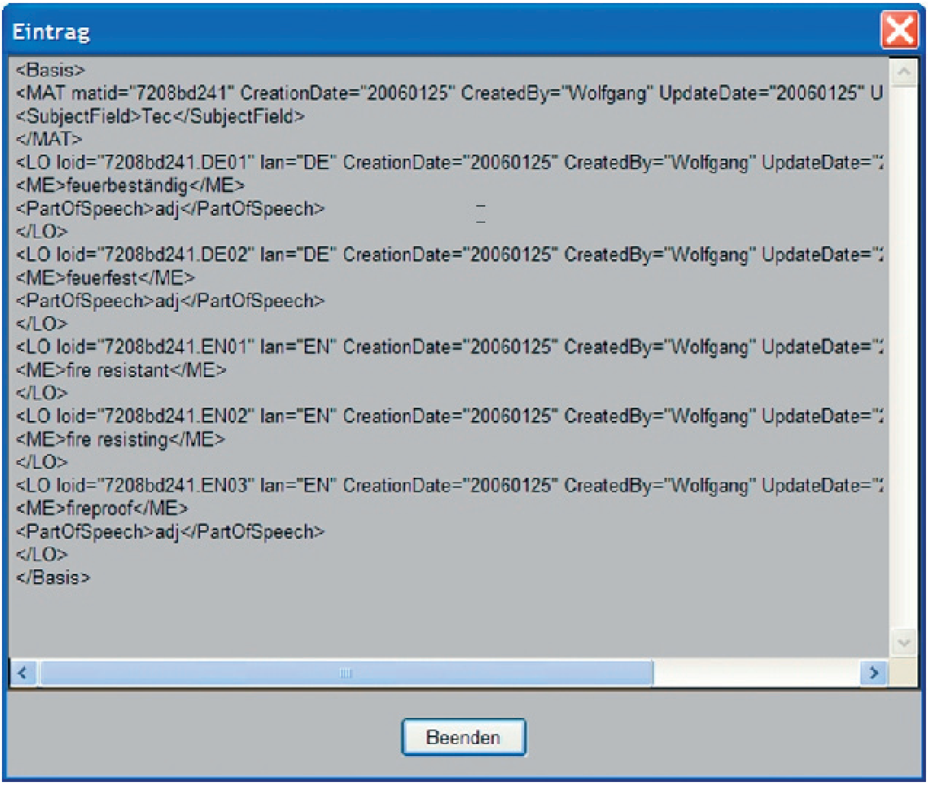
New data fields have been introduced which allow users to perform following operations:

Formatting/layouting within text fields (bold, italic, underline, subscript, superscript).

Inserting cross-references from within a data field to other terms within the database. To manage and control cross-references, a full-fledged link management has been introduced.

Additionally, some further data fields have been indexed to speed up searching and allow switching of the register window to these indexed fields.

3.2 UniTerm Database Organisation



```
<Basis>
<MAT matid="7208bd241" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="20060125" U
<SubjectField>Tec</SubjectField>
</MAT>
<LO loid="7208bd241.DE01" lan="DE" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="
<ME>feuerbeständig</ME>
<PartOfSpeech>adj</PartOfSpeech>
</LO>
<LO loid="7208bd241.DE02" lan="DE" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="
<ME>feuerfest</ME>
<PartOfSpeech>adj</PartOfSpeech>
</LO>
<LO loid="7208bd241.EN01" lan="EN" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="
<ME>fire resistant</ME>
</LO>
<LO loid="7208bd241.EN02" lan="EN" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="
<ME>fire resisting</ME>
</LO>
<LO loid="7208bd241.EN03" lan="EN" CreationDate="20060125" CreatedBy="Wolfgang" UpdateDate="
<ME>fireproof</ME>
<PartOfSpeech>adj</PartOfSpeech>
</LO>
</Basis>
```

Fig. 3: XML representation of a sample coding in the UniTerm database

The UniTerm database is organized as a single-user database that saves XML encoded data in Unicode format. This means that the database allows parallel look-up, but not parallel editing of one database by multiple authors. All entries are automatically XML encoded and saved as XML in the database. The XML format implementation provides some but not all the flexibility of SGML/XML Document Type Definitions (DTD). In UniTerm, all entries are coded in the Unicode UCS2 standard. The database representation of a coding sample is illustrated in Figure 3. This core model structures contains following information:

<Basis> – the multilingual entry.

<MAT> – the common, or language-independent section of the entry which contains concept-based information, e.g. **<SubjectField>**.

<LO loid".." lan=".."> – LO stands for linguistic object. This level is the language section. The language is specified in the *lan* attribute. The second attribute *loid* enumerates multiple language sections within one language and links at the same time the common section of the entry with any number of language sections.

<ME> – main entry, the term.

4 UniTerm and Terminology Exchange

4.1 UniTerm Exchange formats

Generally, UniTerm allows to import and export terminologies. UniTerm supports following **import formats**:

CSV (= comma-separated value list).
XML The XML structure has to be compliant to the UniTerm XML database structure.

For exchange with other applications, users can always choose whether to export a full terminology database or only a selection of it. UniTerm provides data for other applications and TMS in following export formats.

RTF (= Rich Text Format), e.g. for integration into word processors.
UniLex and **UniLex IDS** dictionary. UniLex is the Acolada dictionary range. This export format creates databases in a custom layout to be integrated into the UniLex dictionary range. Standard dictionaries and terminologies are thus integrated for common usage in one system.
Text The text export is a highly flexible export format since users may not only define which data fields and which languages to export but also define text strings preceding and following a data field value and define separators to insert. Examples for text export are a comma-separated value list (**CSV** list) and also a custom XML format which can be directly integrated in other TMS.
HTML (Hypertext mark-up language) which allows easy integration into websites.
UniTerm The UniTerm format is listed here since UniTerm provides sophisticated split / merge functions that allow easy integration of different UniTerm databases into one.
XML This option either allows to export all languages and all entry information or only parts of it. Furthermore, a DTD is automatically generated for the exported XML data to allow validation in XML environments and easy transition process to other XML formats.

Most important for the interoperability with other TMS is the XML import/export function since all relevant terminology standards are formally represented in SGML or XML DTDs. Therefore, the following section provides a sort of checklist which lists potential stumbling blocks for terminology exchange. These difficulties have to be taken into consideration when the exchange of UniTerm data with other TMS is envisaged.

4.2 Problems of Terminology Exchange

Terminology exchange is closely related to standardized terminology formats. In general, standardized formats are intended to facilitate terminology exchange, i.e. to enhance the interoperability between TMS of different vendors. The ultimate goal is terminology exchange without prior negotiation (blind interchange). Blind interchange does not only apply to names of terminological categories but also to values {masc vs. masculine vs. m.} of such categories. Blind interchange also applies to the order of elements which is relevant to most database models. The most widely accepted standards for terminology exchange are:

ISO 12200:2000 MARTIF (= **MA**chine **R**eadable **T**erminology **I**nterchange **F**ormat)
GENETER (= **GENE**ric model for **TER**minology)
OLIF (= **O**pen **L**exicon **I**nterchange **F**ormat)
TBX (= Term**B**ase e**X**change)
TMF (= **T**erminology **M**arkup **F**ramework)

For more information about terminology standards and standardization, see also *http://xml.coverpages.org/terminology.html*.

When exchanging terminologies in XML format, blind interchange will not be possible in most cases for one of the following reasons:

Database restrictions: the data to be imported do not comply with restrictions that the database imposes on data sets. Example of such restrictions are limited size of data fields or of entries.

Different entry models: The entry models of different databases differ with respect to following properties:

a) **Core structure:** the concept models cannot be matched, e.g. one TMS contains one definition per language on the concept level whereas another TMS includes the definition on the term level.

b) **Conflicting element and attribute names:** For example, tags such as <context>…</context> compared to <descripGrp> <descrip type= "Kontext">…</descrip> </descripGrp>.

c) **Mixed content models**, i.e. further tagging (e.g. cross-references, subscript, superscript, layout information) within a data field is not supported or is only supported by different tagging in another TMS.

d) **Conflicting element values:** TMS use different values for the same data category, i.e. the data category *grammaticalGender* has values such as *m.* versus *masc* versus *masculine*.

Different encoding: is the encoding ANSI, Unicode or other? Even Unicode offers different encoding standards, e.g. UTF-8, UTF-16, UCS-2, etc. Transformation from one encoding to another may require additional tools.

The succession of elements does not allow immediate import. The database approach usually does not offer a free succession of elements but defines a fixed order of data fields for import/export. For example, TMS 1 will export *term, context, example* whereas TMS 2 exports the same data fields in the order *term, example, context.*

The XML export formats of both TMS 1 and TMS 2 may create valid instances with regard to a standardized terminology interchange format. The terminology standard – formulated in a DTD – offers more flexibility than the implementation of the standard in the more rigid database approach. The XML-based exports of TMS 1 and TMS 2 can therefore be seen as subsets of the permissible instances defined by the terminology standard itself.

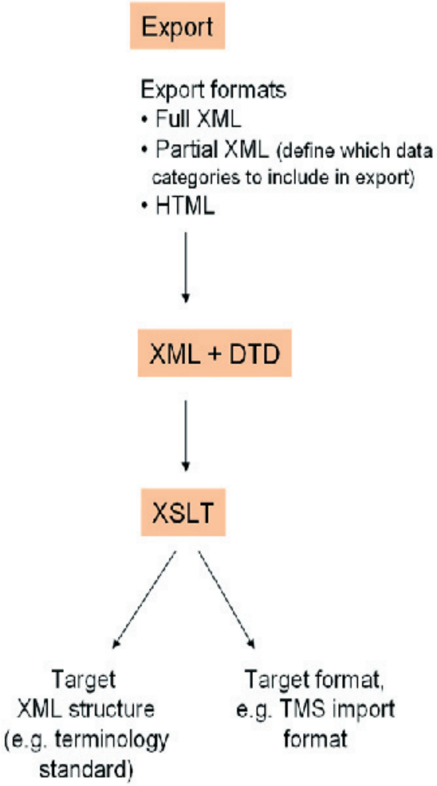

Export
Export formats
• Full XML
• Partial XML (define which data categories to include in export)
• HTML
XML + DTD
XSLT
Target XML structure (e.g. terminology standard)
Target format, e.g. TMS import format

Fig. 4: Model terminology exchange process from XML format

As a consequence, blind interchange will rarely be possible. Instead, interchange needs to be "negotiated", resulting in the implementation of a transformation rule which adapts the export file of TMS 1 to the import structure of TMS 2.

The conclusion is that as long as terminology management systems implement only a subset of permissible instances of a terminology standard into a database but not the standard itself, blind interchange will not be possible.

4.3 Exchanging UniTerm Entries with other TMS

The recommended way of exchanging UniTerm terminologies with other TMS is via the XML export format. UniTerm exports data in an XML format that is similar to the XML format used in the UniTerm database.

Although data categories from ISO 12620 are used in the UniTerm format, the UniTerm XML export does not entirely comply with one of the terminology standards. This means a transformation will be required in most cases if UniTerm XML export data are to be imported into another TMS or a terminology standard. Since the UniTerm core structure very much complies with terminology standards, this transformation is fairly straightforward in most cases if tools like XSLT are used. This strategy, which is illustrated in Figure 4, also enables an automatic exchange between UniTerm and other TMS.

XML export data is provided together with a DTD. The DTD allows to validate the export data in XML environments and speeds up the transformation/integration process.

5 Requirements for Better Terminology Exchange and the UniTerm Enterprise TMS

A number of reasons where terminology exchange is likely to cause problems has been given in the checklist in Section 4.2. With regard to terminological structures, two paradigms seem to conflict: the database approach that current TMS follow and the DTD/schema-based standards for terminology.

Why do different TMS vendors not make sure that the exchange formats created with their systems (and which may even be compliant with a terminology standard) can actually be interchanged? The answer is very simple: terminology exchange is not the primary goal of a TMS. The TMS is built in order to be integrated into a process: integration with a translation memory system, integration with a machine translation system, integration with dictionaries, etc.

Process integration is also the goal of the new UniTerm Enterprise system by Acolada whose first version will be launched in spring 2006. Unlike other TMS, UniTerm Enterprise does not only target translation and localization processes. UniTerm Enterprise is integrated already in the (source language) documentation process and in other processes of internal and external communication. Terminology management thus starts at the source where terms are introduced into a document.

More important for the exchange aspect is that UniTerm Enterprise is the first TMS whose structures are based on DTDs. This means that any standardized DTD for terminology exchange (e.g. SGML DTDs such as MARTIF or XML DTDs such as TBX) can be integrated.

UniTerm Enterprise's default DTD is a concept-oriented custom DTD which has been developed along existing terminology standards and coding practise for structured data. Coding practise favours data categories to be reflected in element names (UniTerm Enterprise) rather than in attribute values (terminology standards).

A number of other categories and information foreseen in standards (transaction information, version history) are fully provided by the UniTerm Enterprise system, i.e. some of the coding is replaced by system functionality. The advantage is that users can actually make use of this information: UniTerm Enterprise offers a full-fledged version management that allows comparison of entry versions and a roll-back mechanism to set back to any previous version.

Additional modules – workflow management and asset management – make UniTerm Enterprise a management system for all terminologically relevant languages resources and the first TMS ever to allow full integration of and working with existing standards for terminology interchange.

6 Conclusion

At present, terminology management systems and standards for terminology exchange follow different paradigms. However, a number of common points and the respect TMS vendors have paid to existing standards when implementing their TMS make negotiated interchange of terminological data an almost trivial task.

TBX as a promoted and widely respected standard for terminology exchange has all chances to become more than an exchange format. With more TMS like UniTerm Enterprise with DTD/schema support actual coding/working with TBX can become more than a vision but common practise.

References

LISA (Localization Industry Standards Association): "Example TBX Conversions". *http://www.lisa.org/standards/tbx/samples [02.02.2006].*

Melby, A. K. (2003): "Interchange using TBX". LISA / OSCAR Meeting, 2003. *http://www.lisa.org/sigs/terminology/tbx_intro/tbx_files/v3_document.htm [02.02.2006].*

Schmitz, K.-D. (1999): "Austausch terminologischer Daten". In: Technische Dokumentation 2. *http://www.doku.net/artikel/austauscht.htm [02.02.2006].*

Endnotes

[1] EURAMIS stands for European Advanced Multilingual Information System.

[2] ISO 12620:1999:"Computer applications in terminology – Data categories".

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:

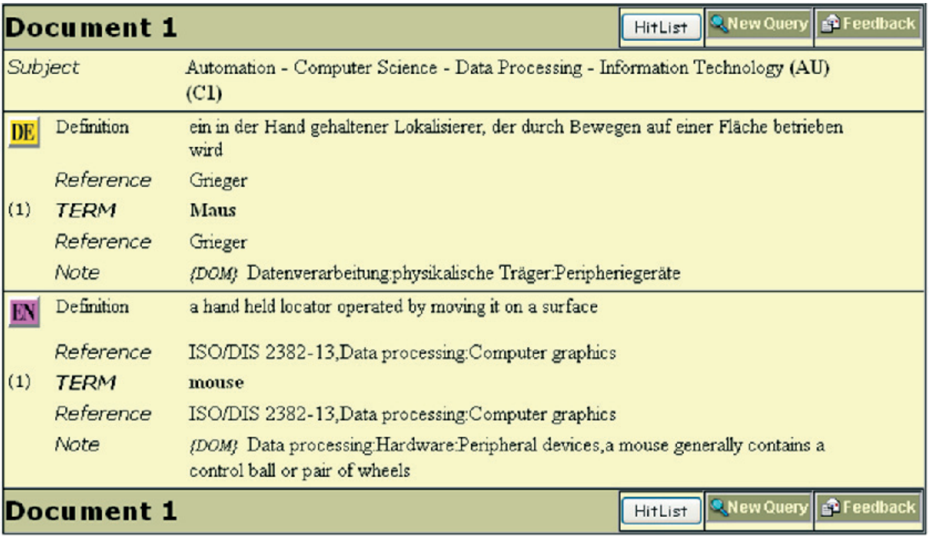
```
CAN “Maus”
CAT NST
ALO “Maus”
TYN (ANI C-POT)
```

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

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

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*).

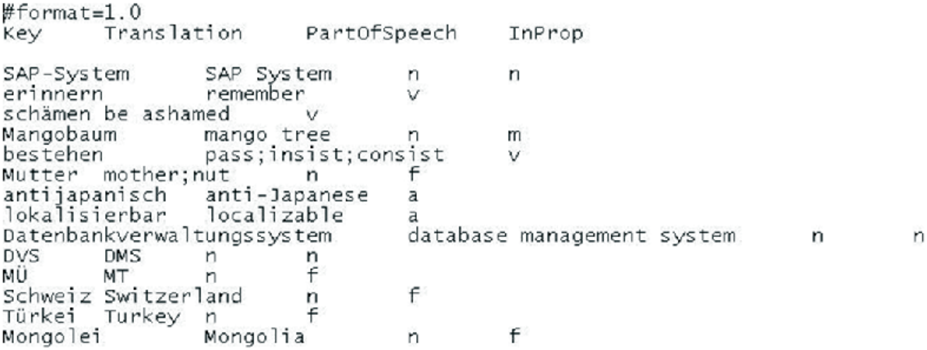
Cross-reference data define semantic relations between the given entry and other entries such as hyponymy, synonymy or meronymy.

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
```

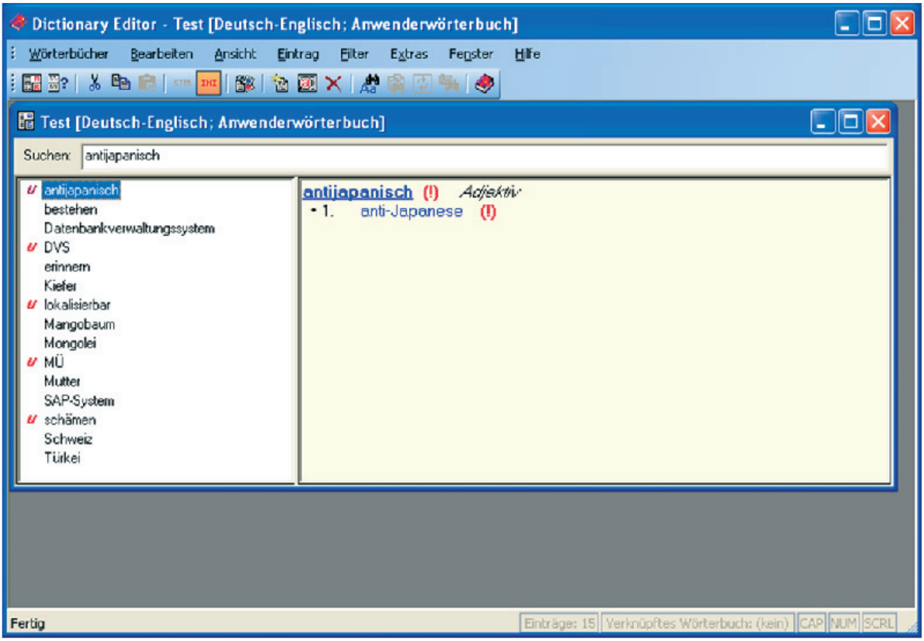
Fig. 2: Promt import format

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



Dictionary Editor - Test [Deutsch-Englisch; Anwenderwörterbuch]

Test [Deutsch-Englisch; Anwenderwörterbuch]

Suchen: antijapanisch

- antijapanisch
- bestehen
- Datenbankverwaltungssystem
- DVS
- erinnern
- Kiefer
- lokalisierbar
- Mangobaum
- Mongolei
- MÜ
- Mutter
- SAP-System
- schämen
- Schweiz
- Türkei

antijapanisch (!) Adjektiv
• 1. anti-Japanese (!)

Fertig

Einträge: 15 | Verknüpftes Wörterbuch: (kein) | CAP | NUM | SCRL

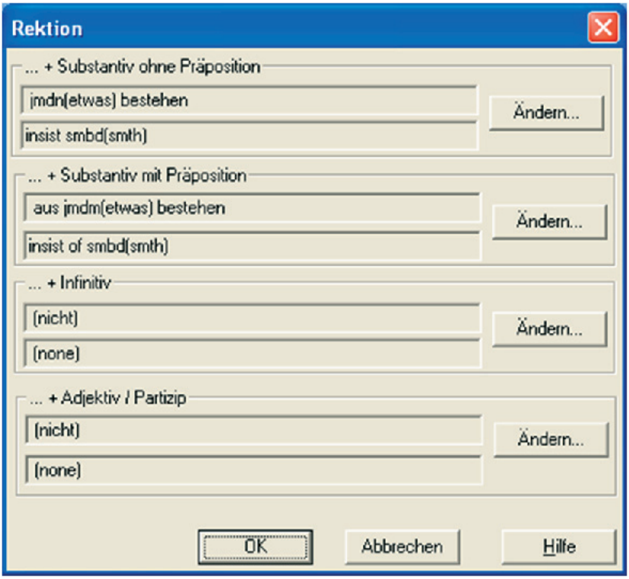
Fig. 3: @Promt Dictionary Editor

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.

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.



Rektion

... + Substantiv ohne Präposition

jmdn(etwas) bestehen

insist smbd(smth)

... + Substantiv mit Präposition

aus jmdm(etwas) bestehen

insist of smbd(smth)

... + Infinitiv

(nicht)

(none)

... + Adjektiv / Partizip

(nicht)

(none)

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

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.

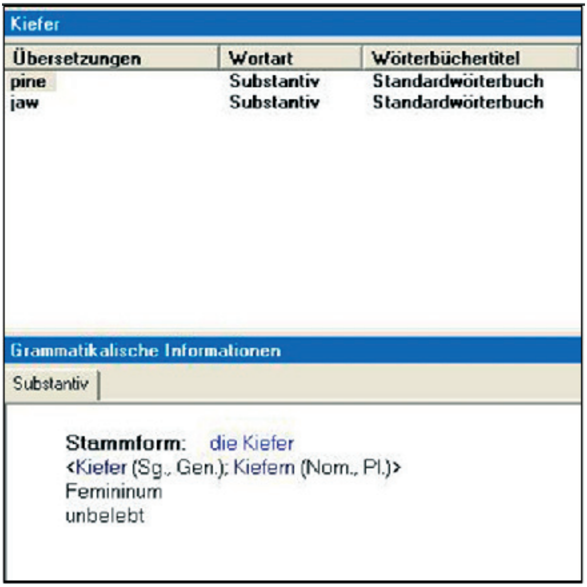
The alternative translations given here are clearly not required as the German source sentences are not ambiguous. This translation error can be explained by the assumption that the different semantic readings of the verb bestehen are internally stored in one entry in the user dictionary which in our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



Kiefer

Übersetzungen	Wortart	Wörterbüchertitel
pine	Substantiv	Standardwörterbuch
jaw	Substantiv	Standardwörterbuch

Grammatikalische Informationen

Substantiv

Stammform: die Kiefer
<Kiefer (Sg., Gen.); Kiefern (Nom., Pl.)>
Femininum
unbelebt

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.

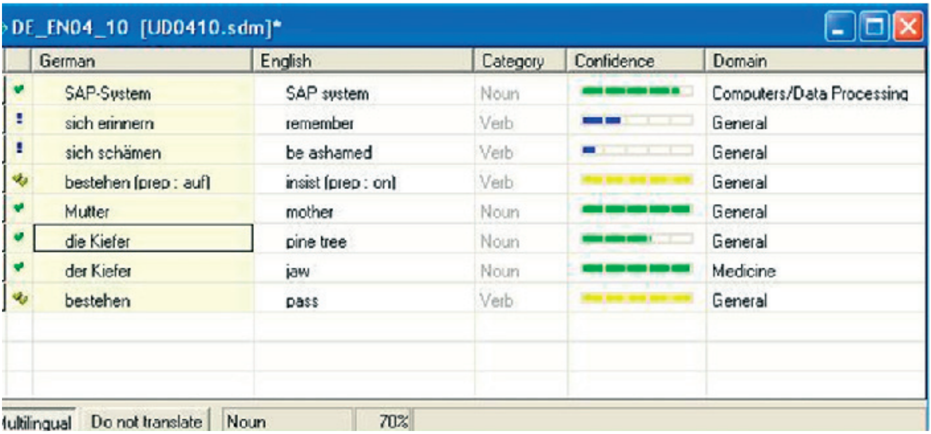
Normalization Dictionaries (NDs) which mainly serve to enhance translation consistency by normalizing SL text before or TL text after translation. NDs help, for example, to avoid orthographic variants, by ensuring that words such as *colour/color* are always spelled in the same way.

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-



DE_EN04_10 [UD0410.sdm]*

	German	English	Category	Confidence	Domain
✓	SAP-System	SAP system	Noun	🟩🟩🟩🟩🟩	Computers/Data Processing
!	sich erinnern	remember	Verb	🟦🟦	General
!	sich schämen	be ashamed	Verb	🟦	General
✓✓	bestehen [prep : auf]	insist [prep : on]	Verb	🟨🟨🟨🟨🟨🟨	General
✓	Mutter	mother	Noun	🟩🟩🟩🟩🟩🟩	General
✓	die Kiefer	pine tree	Noun	🟩🟩🟩	General
✓	der Kiefer	jaw	Noun	🟩🟩🟩🟩🟩🟩	Medicine
✓✓	bestehen	pass	Verb	🟨🟨🟨🟨🟨🟨	General

ultilingual | Do not translate | Noun | 70%

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).

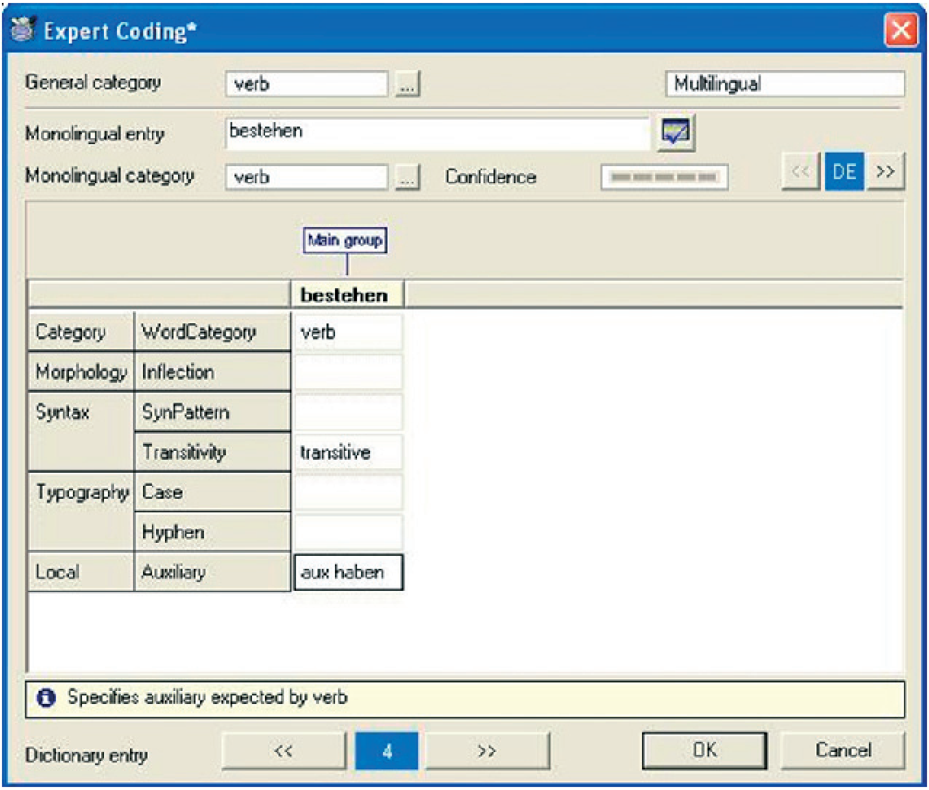
English	German
to light	anzünden
a light	Licht
light	leicht

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



General category: verb

Multilingual

Monolingual entry: bestehen

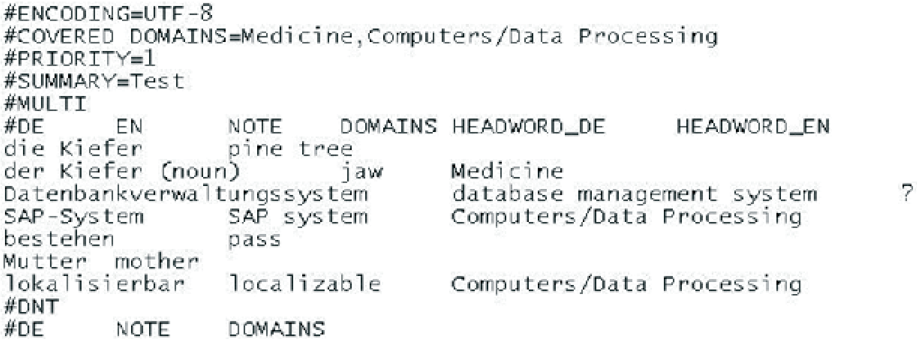
Monolingual category: verb

Confidence

		bestehen
Category	WordCategory	verb
Morphology	Inflection	
Syntax	SynPattern	
	Transitivity	transitive
Typography	Case	
	Hyphen	
Local	Auxiliary	aux haben

Specifies auxiliary expected by verb

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.

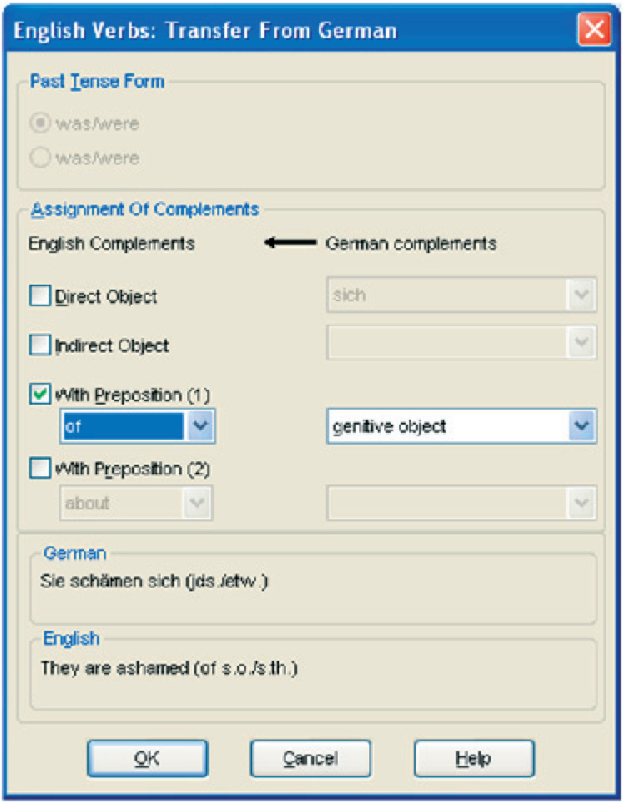
The lexicon exchange with other MT systems is not as straightforward, though. Similar to other systems,
Translate pro offers the possibility to import bilingual glossaries as text files. As these glossaries contain only word pairs and no information on the grammatical category the user is advised to 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



English Verbs: Transfer From German

Past Tense Form

● was/were
○ was/were

Assignment Of Complements

English Complements ← German complements

□ Direct Object — sich
□ Indirect Object
☑ With Preposition (1) — of — genitive object
□ With Preposition (2) — about

German
Sie schämen sich (jds./etw.)

English
They are ashamed (of s.o./s.th.)

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.

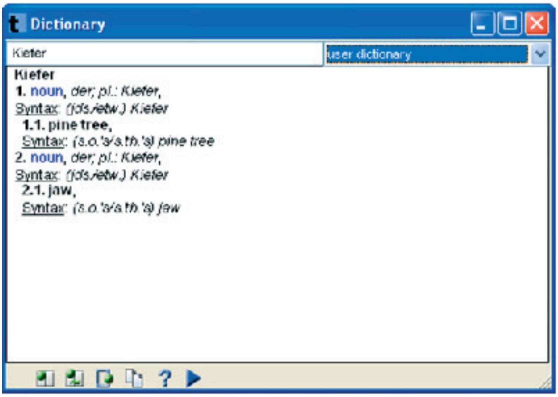
The second possibility consists of coding two distinct entries in the XML file with one `<target>` group each. This solution, which results in creating two noun entries during lexicon import (see Figure 10) is clearly preferable. Although the first noun has wrongly been assigned masculine gender by the 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-



Kiefer

user dictionary

Kiefer
1. noun, *der; pl.: Kiefer,*
Syntax: *(jd's./etw.) Kiefer*
1.1. pine tree,
Syntax: *(s.o.'s/s.th.'s) pine tree*
2. noun, *der; pl.: Kiefer,*
Syntax: *(jd's./etw.) Kiefer*
2.1. jaw,
Syntax: *(s.o.'s/s.th.'s) jaw*

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.

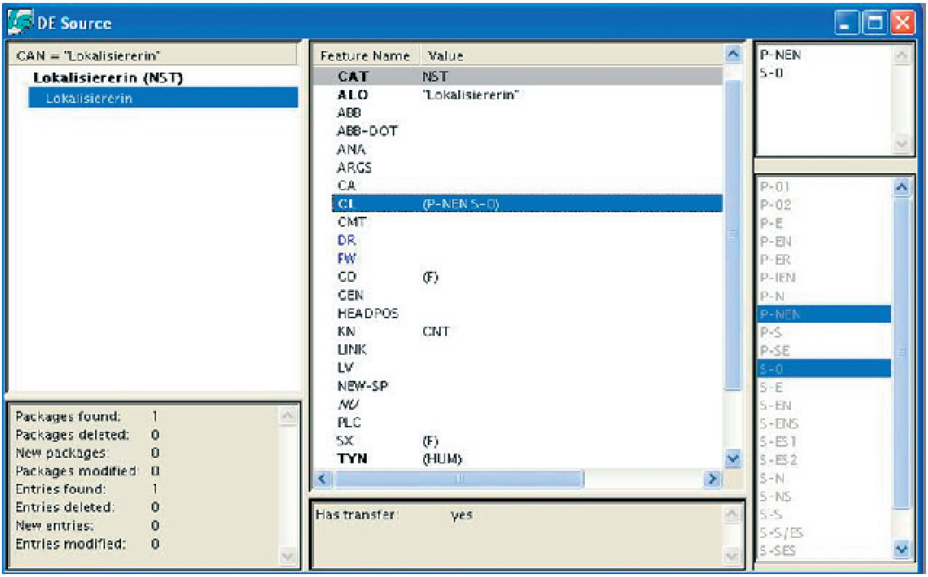
KN (kind of noun): KN is a syntactic-semantic feature which is used to distinguish between mass and count nouns. *Lokalisiererin* takes the value CNT, i.e. this noun is countable.

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



CAN = 'Lokalisiererin'

Lokalisiererin (NST)
Lokalisiererin

Packages found:	1
Packages deleted:	0
New packages:	0
Packages modified:	0
Entries found:	1
Entries deleted:	0
New entries:	0
Entries modified:	0

Feature Name	Value
CAT	NST
ALO	'Lokalisiererin'
ABB	
ABB-DOT	
ANA	
ARGS	
CA	
CL	(P-NEN S-0)
CMT	
DR	
FW	
GD	(F)
GEN	
HEADPOS	
KN	CNT
LINK	
LV	
NEW-SP	
NV	
PLC	
SX	(F)
TYN	(HUM)

Has transfer: yes

P-NEN
S-0

P-01
P-02
P-E
P-EN
P-ER
P-IEN
P-N
P-NEN
P-S
P-SE
S-0
S-E
S-EN
S-ENS
S-ES1
S-ES2
S-N
S-NS
S-S
S-S/ES
S-SES

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.:

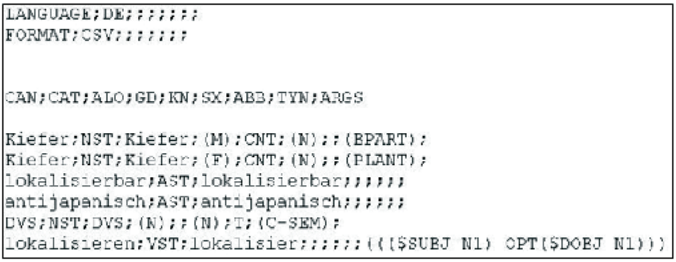
```
:LANGUAGE DE
:FORMAT INTERNAL
(CAN “Brief” CAT NST ALO
“BRIEF” CL (P-E S-S/ES) GD (M)
KN CNT SX (N) TYN (ABS CNC
SEM))
```

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



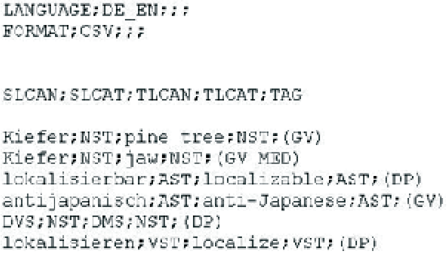
```
LANGUAGE;DE;;;;;;;
FORMAT;CSV;;;;;;;


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 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].



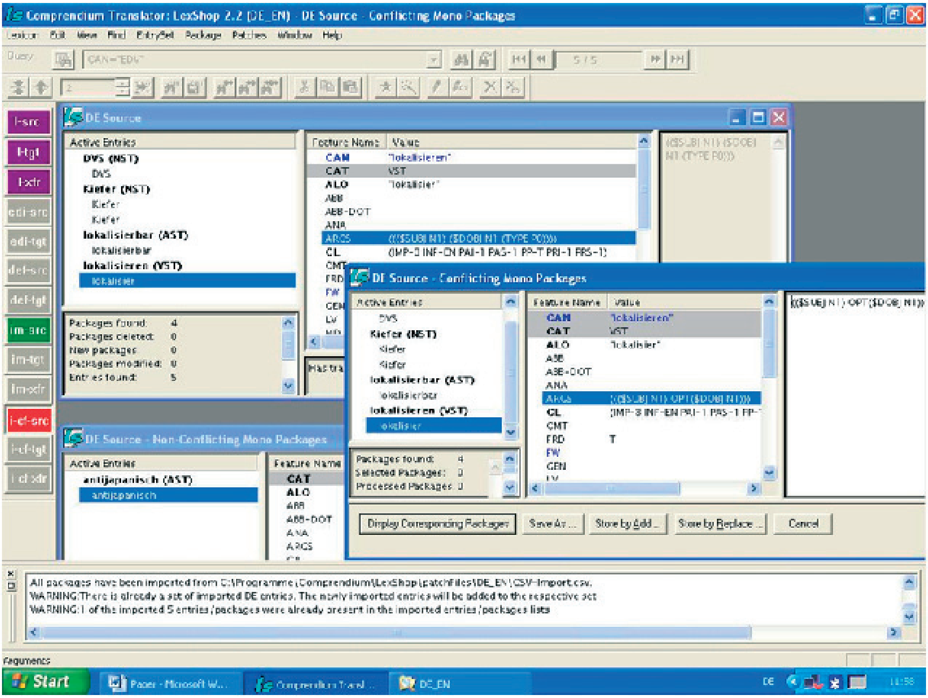
```
LANGUAGE;DE_EN;;;
FORMAT;CSV;;;


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

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-


Comprendium Translator: LexShop 2.2 (DE_EN) - DE Source - Conflicting Mono Packages
Lexicon Edit View Find EntrySet Package Patches Window Help
DE Source
DE Source - Conflicting Mono Packages
DE Source - Non-Conflicting Mono Packages
Active Entries
Feature Name
Value
Display Corresponding Packages
Save As ...
Store by Add ...
Store by Replace ...
Cancel
Start

```
All packages have been imported from C:\Programme\Comprendium\LexShop\patchFiles\DE_EN\CSV-Import.csv.
WARNING:There is already a set of imported DE entries. The newly imported entries will be added to the respective set
WARNING:1 of the imported 5 entries/packages were already present in the imported entries/packages lists
```

Fig. 14: Importing conflicting monopackages in LexShop

lopment Environment (LDE) of the LogOSMaTran 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:

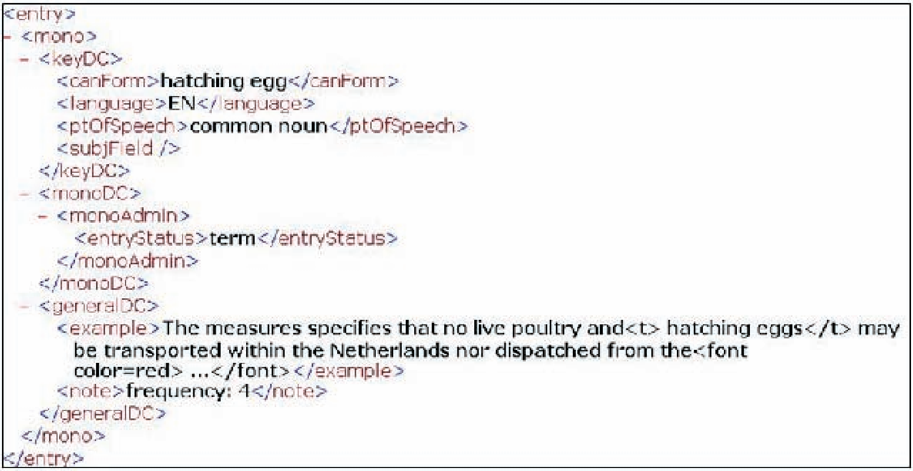
```
Source_Language; Source_
Word; Head_Word; Source_POS;
Source_Gender; Target_Language;
Target_Word; Target_Gender;
Company_Code; Subject_Matter_
Code; Lexicon_Source.
```

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

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

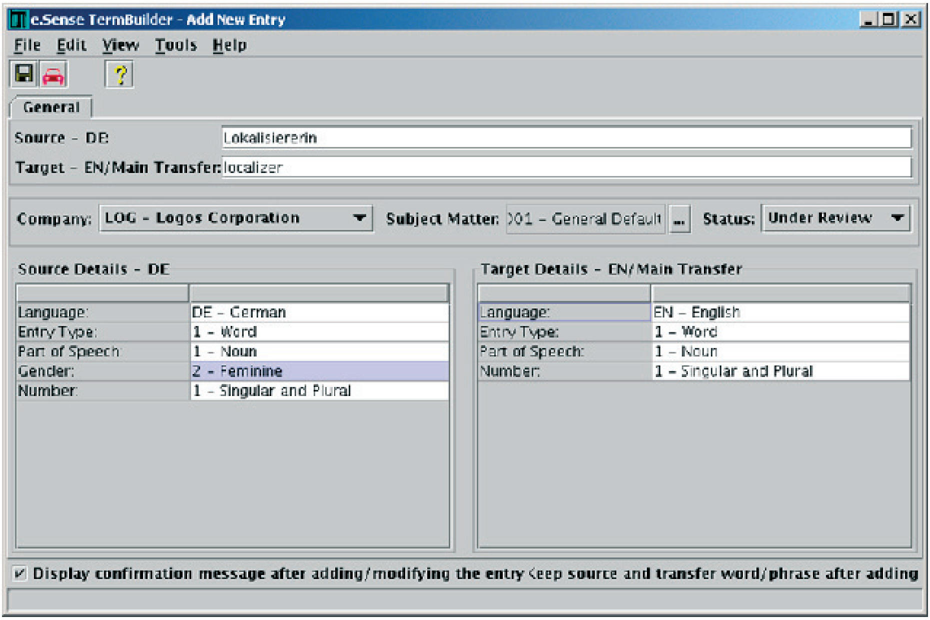
In this example, which is taken from an export file, the value for the target gender is empty.

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



Source - DE: Lokalisiererin

Target - EN/Main Transfer: localizer

Company: LOG - Logos Corporation

Subject Matter:)01 - General Default

Status: Under Review

Source Details - DE

Language:	DE - German
Entry Type:	1 - Word
Part of Speech:	1 - Noun
Gender:	2 - Feminine
Number:	1 - Singular and Plural

Target Details - EN/Main Transfer

Language:	EN - English
Entry Type:	1 - Word
Part of Speech:	1 - Noun
Number:	1 - Singular and Plural

☑ Display confirmation message after adding/modifying the entry ‹eep source and transfer word/phrase after adding

Fig. 16: OpenLogos TermBuilder

by Translate pro (see Section 3.3) and Linguatec.

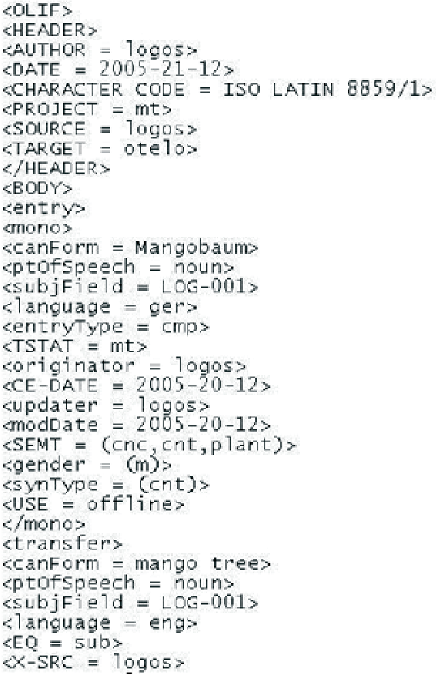
The exported entry for the entry Mangobaum, for example, has the following structure:

```
<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.

Gregor Thurmair

Exchange Formats: TBX, OLIF, and Beyond

Abstract

This paper tries to comment on some of the standardisation efforts in the area of exchange formats for lexical resources. The first family of standards was centred around terminological data, producing exchange formats like MATER/MARTIF and TBX, based on an organisation of the data as concepts and (language-specific) terms. When the exchange of fully annotated lexical data came into play, standards like OLIF and MILE were proposed; they focus on the representation and the exchange of (mono- and multilingual) dictionary entries and their attributes (Thurmair/Lieske 2002). Recent developments are organised around the creation of markup frameworks, try to define frameworks for meta-models on one hand, and sets of elementary data categories on the other hand, both of which can be grouped into workable exchange formats.

1 TBX

1.1 History

The first exchange format for terminology was called MATER; it defined how data had to be stored on a magnetic tape, specifying, among other things, byte sequence, tape length, block size etc. This format was converted into Micro-Mater (for PC exchange), and later into MARTIF, the first SGML-based format. Martif underwent several standardisation steps (ISO 12200, ISO 12620 and others) and was further developed in an EU funded project called SALT. The current status of the format is XLT (XML-based Formats for Lexicon and Terminology Exchange) which is the framework for several flavours of the standard depending on the different use cases; the most widely known format of these is TBX (Term Base eXchange) which is promoted by LISA, the Localisation Industry Standards Association (*www.lisa.org/standards/tbx*).

1.2 Terminological Entry

TBX models a terminological entry. Such entries are built upon the distinction between concepts (which are semantic units) and terms (which designate such units in different languages). One of the first terminological databases, the TEAM system (Hohnhold 1984), consisted of a meta-language header, covering the concept identification, subject area, term status and other general features, and language-specific sections containing the terms, with denotations, part of speech, definitions, and other language-specific material (see Figure 1).

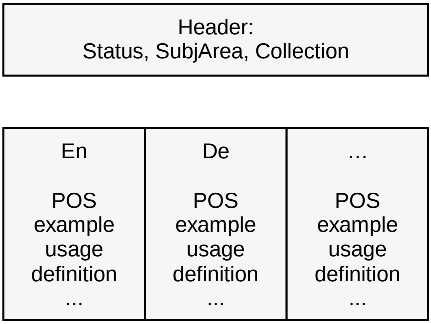


Fig. 1: TEAM database organisation

1.3 TBX Description

The distinction between concepts and terms is still a basic element in the TBX architecture with terminological entries being organised by concepts. Concepts are basic semantic entities; they can have global attributes (stored in the au-

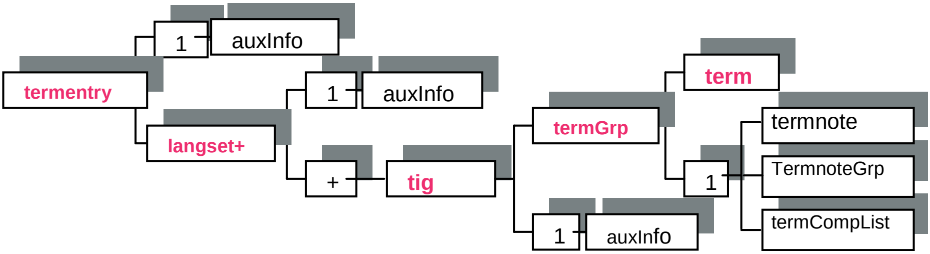


Fig. 2: TBX Term entry structure

xInfo section), like subject area, related concepts, definition, example, sample sentences etc. Then they are described in language-related designation sections (langsets) which consist of term information groups which enclose the single terms. The structure of a term entry is described in Figure 2.

Term entries form the core of a TBX file which is an XML document consisting of the following components (see Figure 3):

A **header** which describes the file by providing some global and administrative information (content, validation status, contact, encoding, revisions, etc.).

A **body** which consists of a set of entries, one per concept in the database. The body may have introductory and concluding elements (see Figure 3).

A sample TBX file, taken from the description in *www.lisa.org/standards/tbx*, is given in Figure 4 below.

1.4 Discussion

The meta-model of a TBX entry provides two characteristics:

1. The basic elements of the exchange are concepts, i.e. groups of terms. TBX is based on the distinction between a **concept** (*'Begriff'*) considered to be a unit of thought constituted through abstraction on the basis of properties common to a set of objects; concepts are not bound to particular languages.), and a **term** (*'Benennung'*) considered to be the designation of a defined concept in a special language by a linguistic expression.). As a result, a TBX entry does not consist of single terms but of sets of terms.

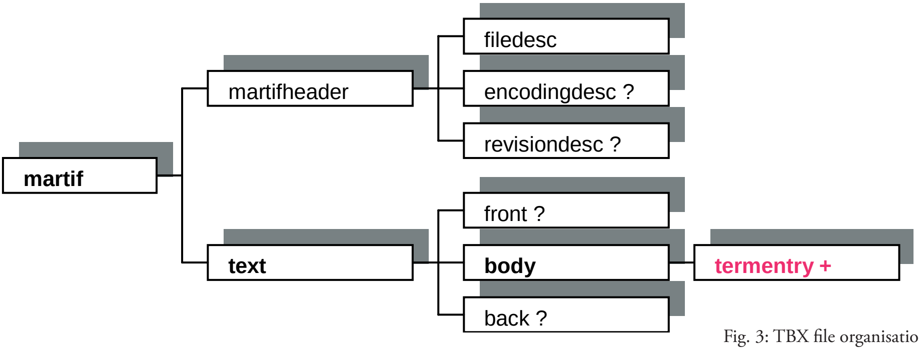


Fig. 3: TBX file organisation

```
<?xml version='1.0'?>
  <!DOCTYPE martif SYSTEM  „./TBXcoreStructureDTD-v-1-0.DTD“>
  <martif type='TBX' xml:lang='en' >
  <martifHeader>
    <fileDesc>
      <sourceDesc><p>from an Oracle corporation term-Base</p></
sourceDesc>
    </fileDesc>
    <encodingDesc><p type='DCSName'>TBXdefaultXCS-v-1-0.XML</p></
encodingDesc>
  </martifHeader>
  <text> <body>
    <termEntry id='eid-Oracle-67'>
      <descrip type='subjectField'>manufacturing</descrip>
      <descrip type='definition'>A value between 0 and 1 used in
…</descrip>
      <langSet xml:ang='en'>
        <tig>
          <term tid='tid-Oracle-67-en1'>alpha smoothing factor</
term>
          <termNote type='termType'>fullForm</termNote>
```

Fig. 4: Example of a TBX file (English and Hungarian terms)

2. As a consequence, the meta-model is **multilingual**, i.e. there are as many languages as equivalents are provided, all of which can be interchanged freely; and it is **non-directed**, i.e. from a German-English-French term base all possible bilingual terms can be extracted: German-English, English-German, French-German, etc.

An example is given in Figure 5 which shows two German equivalents for a French entry; they are assumed to be synonyms, both linked to the French term.

2 From TBX to OLIF

The first annotated linguistic dictionaries to be exchanged were Machine Translation resources. However, when starting to work on an exchange format for Machine Translation dictionaries, it quickly became obvious that MARTIF / TBX was not able to satisfy the requirements for exchanging such dictionaries. This is due to the fact that terminology and dictionary entries follow different conceptual lines (Hohnhold / Schneider 1991); but it also follows from inherent problems of the TBX standard.

When exchanging MT information, there were three basic questions to be solved: What are the units of exchange? How can the annotations of the single units be described? How are the relations between them organised?

2.1 Annotations of Exchange Units

The attempt to exchange monolingual MT dictionaries failed rather quickly.

1. The linguistic descriptions available in the TBX standard were not satisfactory for linguistic exchange. In ISO 12620, only very few

```
<termEntry>
  <langSet lang=‘“fr“>
    <ntig>
      <termGrp>
        <term>échantillonneur</term>
      </termGrp>
    </ntig>
  </langSet>

  <langSet lang=“de“>
    <ntig>
      <termGrp>
        <term>Abtastglied</term>
      </termGrp>
    </ntig>
```

Fig. 5: French term with two German synonyms

annotations are covered (like part of speech, gender, number), and they refer to very few languages. In particular, there was no notion of the basic features to be exchanged in MT, like inflection paradigm, syntactic types, argument structures, semantic features etc., not even complete part of speech sets were provided[1]. As a result, it became clear quite quickly that an extension was required to cover most of the features which an MT system was supposed to exchange.

2. The organisation of the linguistic annotations was not obvious. Some were linked to the concept level (like definitions, examples, relations like broader/narrower term), others, like part of speech (morpho-syntax) or animacy (semantics) were linked to the term level. As a result, semantic information is represented both on concept and on denotation level which is not intuitive. Therefore, it became necessary to define the basic annotations (attributes and their legal values) for the linguistic information to be exchanged. Previous work (e.g. in EAGLES), as well as inspection of existing MT dictionaries, could be used as a reference.

2.2 Relations between units

The attempt to exchange transfer MT dictionaries failed rather quickly as well.

Most MT systems disambiguate 1:n transfers by **tests and actions**, which is shown in Figure 6[2]:

Transfer entries describe language-pair-specific relations between concepts. As TBX is intended to be multilingual and non-directed, there is no possibility to define bi-language directed information as transfer tests. TBX does not provide means to attach information to the links between the denotations in different languages; there are just term information groups relating information to a monolingual term, and information for concepts; options to further qualify

de *ausführen*	(if direct object is of type <person>)	-> en *take out*
de *ausführen*	(if direct object is of type <program>)	-> en *execute*

Fig. 6: Example of a MT transfer entry

[concept: <kill>]		
de *töten*	(standard language)	-> en *kill*
de *umlegen*	(slang)	-> en *bump off*

Fig. 7: Language register

the relations between concepts and denotations, or denotations of different languages do not exist.

This fact does not just relate to MT transfer tests but also to other phenomena, like language register where the same concept has different denotations depending on the register chosen (see Figure 7).

The same holds for other kinds of relations between concepts, like the ones defined in the ISO 2788 for monolingual or multilingual thesauri, or the more elaborate ones as defined in EuroWordNet (Vossen 1999), only limited relations are defined in TBX.

In general, the TBX model assumes that all denotations of a langset are synonyms, and all langsets are equivalents (as shown in the synonym example above, see Figure 5); there is no possibility to qualify such relations in any way.

There is another consequence of a multilingual non-directed approach: In theory it should be possible to **revert** transfers and create an arbitrary bilingual dictionary from such lists. However, this has never worked in practice. Several attempts to create e.g. an English-French dictionary from a German-English-French source failed. As a matter of fact, authors start writing in their native language, and search for equivalents in other languages, which means that such terminology entries are de facto directed, and cannot simply be reverted. Very often, the target equivalents are a bit more general than the source term, (e.g. de *Lichtbogen* -> en *arc*); this fact results in a very specific and improper translation for a rather general term if the entry is reverted.

As a result, in OLIF a data category called `<equival>` was introduced in the transfer section which encodes the degree of equivalence between two words or phrases. Its value indicates whether an entry can be reverted or not. For MT dictionary exchange it is necessary to model the relations between the members of a TBX langset explicitly, moving the basic unit of exchange from a non-lingual set of terms to a monolingual concept/term.

2.3 Units of Exchange

As there is no general mechanism of linking particular source and target terms, there is no means to define equivalents for **general vocabulary** expressions. As such words, like *find, search, restriction* etc. are quite ambiguous, and need to be defined in the context of the respective language, they cannot be stated in a concept – term type manner.

Therefore, TBX cannot be used for exchange of large portions of MT dictionary terms as the majority of MT dictionary entries are general vocabulary terms. TBX does not claim to support general vocabulary terms, and states that the exchange format is intended to support terminology only.

However, the question is which theoretical distinction underlies the fraction of language that is covered by TBX. While it is supposed to cover terminology (as opposed to general language), this does not seem to be the case: Terminology in areas which are subject to societal or cultural influences is not covered either: In the area of the educational system, legal system, social welfare etc., there is no (non-lingual) concept with terms in many languages; very often there is not even a translation available as the underlying phenomenon does not exist in other societies or languages, although the concepts are clearly special-language terms, and match all requirements of being a term (like the Irish *Leaving certificate*, the German *Abitur* or the English *solicitor*). TBX is suitable for the representation of technical terms where a 1:1 correspondence between participating languages can be assumed.

As a result, TBX is not able to support the exchange requirements of linguistic resources, be it for machine translation, for monolingual applications, for WorldNet type conceptual relations, or any other linguistic tasks. It only covers a part of terminology exchange.

The reason for this fact lies in a **conceptual inadequacy** of the terminological approach: It assumes that there is a concept which has designations in different languages. This idea separates a concept from language, and in turn makes the concept itself a non-language phenomenon, which is not the case: Following Hegel (1807), a concept can only be thought of in the form of a language expression. It is a commonplace since Saussure's *Cours de linguistique générale* that language is a system of signs, and the meaning of a sign is at least partially co-determined by its position in the language system: The consequence is that a non-lingual concept, without being related to other signs of a language system, cannot be defined.

This is the reason why TBX cannot define general purpose words, nor terminology which is defined in language or social specific contexts. This is also the reason why TBX cannot express relations between the terms of two languages, nor assign linguistic descriptions to concepts.

The conclusion is that concepts are monolingual linguistic entities, and must be described in monolingual terms. The consequence is, then, that there must be an explicit relationship between a monolingual entity in one language, and monolingual entities in other languages.

This is the approach which was taken by OLIF: Concepts are monolingual entities, and relations between concepts are modelled explicitly. This approach is bi- or multilingual, and directed. It is more general than the TBX approach, in fact, the relationship of full and reversible equivalents assumed by TBX covers just one specific case of how such a conceptual relationship can be described.

3 The OLIF Format

3.1 History

The Open Lexicon Interchange Format (OLIF) was first defined in an EC project called OTELO. It was intended to enable OTELO partners to exchange sets of MT entries between MT vendors and MT users; one of the objectives was to provide term data (from a term base like SAPterm) for use in MT systems such as Logos or METAL; it included the exchange format itself as well as converters provided by the MT vendors from and into OLIF.

Later versions of the exchange format were developed by the OLIF consortium, members of which included the main MT providers (Systran, Logos, SailLabs, linguatec) and terminology providers and users (Trados, Microsoft, IBM, European Commission, and others). The initative was (and still is) headed by SAP. The current version added a header structure like TBX, provisions for multilingual ontologies, better XML structuring, and several tools and supportive components (McCormick/Lieske 2005).

OLIF is used by major MT users like the European Commission, European Patent Office, SAP, and other multi-vendor MT systems.

As opposed to other standards like EAGLES/MILE (Calzolari et al. 2002), OLIF intended to be pragmatic, and only exchange information which existing MT dictionaries provide, or can make use of. No information which is not (yet) in use, or which is idiosyncratic to a particular system should be included in the standard.

3.2 The OLIF Meta-model

The basic architecture decision of OLIF was to be concept-based (i.e. the basic unit is a semantic entity); but different from TBX, concepts in OLIF are defined for a given language. Concepts form the nodes of an OLIF entry. Between concepts, there are links which point from one concept to another; these links can be monolingual (in case of thesaurus relations) or multilingual (in case of translations). As a result, the metamodel of OLIF can be characterised as follows:

1. It is concept-based but concepts are monolingual and have linguistic annotations.
2. It is multilingual (there can be links from a concept to many target language nodes) but directed (the links have a source and a target, and cannot easily be reverted).

3.3 The OLIF Entries

3.3.1 Key Description

The first challenge is to characterise the entries of exchange. OLIF entries (see Figures 8 and 9) are characterised by four types of information: a canonical form, a language, a part of speech, and a semantic tag[3].

The **canonical form** needs to be described in more detail, to answer questions like Beamter vs. Beamte, automatischer Anlasser vs. automatische Anlasser vs. automatisch Anlasser (multiword terms in particular can be found in many variants).

The **language** is the language in which the entry is defined.

The **part of speech** was defined based on the EAGLES recommendations; in OLIF only open word classes are supposed to be exchanged (as most of the MT systems have their own idiosyncratic view to function words), so noun, verb, adjective and adverb are the categories used. In case

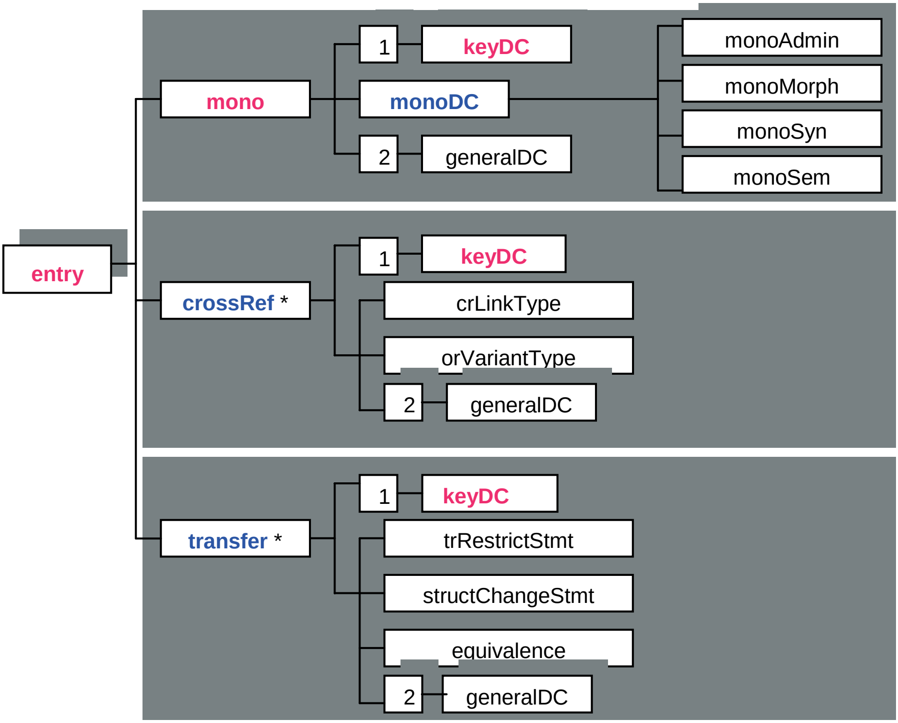


Fig. 8: Structure of an OLIF entry

```
<entry>
  <mono>
    <keyDC>
      <canForm>table</canForm>
      <language>en</language>
      <ptOfSpeech>noun</ptOfSpeech>
      <subjField>general</subjField>
      <semReading>86</semReading>
    </keyDC>
    <monoDC>
      <monoMorph>
        <inflection>like book,books</inflection>
      </monoMorph>
      <monoSyn>
        <synType>cnt</synType>
        <synFrame>[gencomp-opt]</synFrame>
      </monoSyn>
      <monoSem>
        <semType>inform</semType>
      </monoSem>
    </monoDC>
  </mono>
  <crossRefer>
    <keyDC>
      <canForm>row</canForm>
      <language>en</language>
      <ptOfSpeech>noun</ptOfSpeech>
      <subjField>general</subjField>
      <semReading>69</semReading>
    </keyDC>
```

Fig. 9: Example of OLIF entry (from MCCORMICK / LIESKE 2005)

closed classes need to be exchanged, the EAGLES definitions can be used.

The **semantic tag** provides difficulty. In practical situations, the only semantic description available is the **subject field tag**. This can be used as a means for disambiguation. However, there can be cases where the same concept can belong to several subject fields (hand grenade can support both Military and Law-Police subject fields), and several concepts can belong to the same subject fields (like *key* in IT: *Code*, *Schlüssel, Taste*, etc.). Therefore a more precise description had to be found, and a **reading number** was used in addition[4].

3.3.2 Monolingual Annotations

The entry nodes can have linguistic annotations. Such annotations refer to the linguistic and terminological items to be exchanged, and can be grouped according to the levels of linguistic descriptions:

Morphological information is included in most MT lexicons, albeit in very different form as far as inflection information is concerned: While some systems provide different entries for different stems of an inflection paradigm, others enumerate inflection classes for a given stem. As nearly all systems have mechanisms to default the inflection paradigm for unknown words, the idea was to use such components, and use an example-based approach in the exchange standard: Inflection classes are given as simple words, used as examples (`inflects_like`). So, inflection classes in OLIF are Bach, Auto, Haus etc., and it is left to the participating systems to generate their respective internal information structures.

Syntactic information: OLIF provides two basic information types: Syntactic type, i.e. some subcategorisation of the main parts of speech (like the distinction between mass vs. count nouns), and syntactic frames which specify the syntactic argument structure of the respective entries. In particular the argument structure is coded very differently in different MT system, and needs to be converted by each of them from and into OLIF.

Semantic information is also coded. An analysis of the existing MT systems showed that most of them use a simple type system with values such as *human*, *animate*, *place* etc.; more elaborate information is not specified in OLIF as it is not available in such systems.

There is also a section with **administrative information** where author, last editor, validation status and other information is stored.

In general, the objective of OLIF is to cover all such features which make sense to be exchanged. In addition, every system has its own internal information (like: hyphenation information; location in various system dictionaries and the like); this was considered idiosyncratic and did not become part of the standard. Also, information which is relevant but not existing in most systems (like elaborate semantic descriptions) is not part of the standard.

3.3.3 Links

Entries can be connected by links. There are two basic types of links: Links which combine monolingual entries (cross-references), and links combining entries of different languages (translations). Links are directed, i.e. they lead from a source entry (characterised by a key description) to a target entry (characterised by another key description). In addition, links can have attributes:

Crossreferences have a link type (e.g. `is_broader_term_of`, `has_meronym` etc.). The link types have been derived from EuroWordNet; the idea is to support resources used in retrieval, e.g. for query expansion.

Translations have more complex link annotations, consisting mainly of structural descriptions, defining a syntactic configuration (in form of underspecified trees) which must be satisfied for a given link to be activated (i.e. definitions of transfer tests), and
structural changes, defining constellations which define target language changes to be triggered for certain transfers.

It can easily be seen that such attributes of links require them to be directed: Attributes for a German-English link differ significantly from attributes for an English-German entry, even if the same entries are involved. A model like TBX covers only a special case (no link attributes given), and only in this special case a link is reversible.

3.4 OLIF as an Exchange Format

The concept of an exchange format reflects the fact that different systems use different internal representations for dictionary material: For instance, some distinguish between single word and multiword dictionaries, others don’t.

As a result, each system participating in the exchange must provide converters from and into OLIF, whereby the proprietary format is converted into the exchange format. Such converters face a number of challenges, given the requirement that conversions should be fully automatic, and complete, i.e. the conversion of a dictionary entry file into OLIF and back should result 1:1 in the same dictionary file:

As OLIF only covers the parts of MT entries which make sense to exchange, there are always **idiosyncratic** parts of the dictionary which are not part of the standard. To be able to exchange the complete dictionary therefore requires system-specific extensions to the standard.

The converters have to cope with all kinds of **mismatches** between the MT systems and the standard definitions:

Proper names are treated as special part of speech in one system and as syntactic subcategorisations in another; mass nouns are considered to be semantic in one system, syntactic in another one. This type of mismatch requires recomputation of the respective values in the converters.

Morphology is a particularly tricky area: Some systems store alternative word stems (for umlauts, irregular verb forms etc.) while the standard only gives numbers of classes, represented by examples. The converters must reconstruct the appropriate information structures when importing OLIF entries.

The **key descriptions** create overhead if dictionary entries are exported and imported for the same system; in this case, the system-internal ID numbers would stay valid, and could easily be used as unique definition marker for a given lexical unit.

The biggest problem in writing converters, however, turned out to be the concept-based **organisation** of OLIF. Most MT systems are lemma-based, and tend to conflate different concepts into one entry to avoid the creation of ambiguities in analysis and parsing. As a result, a syntactic frame like `<Subject – optional Direct_Object>` could describe one concept (with an optional direct object), or two of them (an intransitive one, and a transitive one). If such a concept then has three translations, it is hard to see how the correct assignment of lexical units (with their syntactic description) to translation links could be achieved in a fully automatic way. More research is required to make progress in this area and find a dictionary organisation which keeps the concept-based orientation without giving up the processing advantages of the lemma-based approach.

As a result, writing programs which convert dictionaries from and into OLIF fully automatically and without loss of information is a challenging task.

4 Beyond OLIF

Since OLIF was defined, several other standardisation proposals have been discussed.

4.1. MILE

The MILE standard is the result of research based on EAGLES / PAROLE (Atkins et al. 2002, Ide et al. 2003). It presents the representation of multilingual information in the framework of a layered lexicon representation standard; the morphosyntax being defined by PAROLE, the semantics by SIMPLE, and the multilinguality by ISLE. Unlike OLIF, MILE covers not just the information items which are available in today's MT lexicons but intends to present a complete lexical description, including semantic representation and multilinguality.

In fact, MILE is not an exchange standard but a representation standard, and can be mapped into several different exchange formats as long as they have the expressive power to support all the MILE information categories which holds neither for TBX nor for OLIF.

As it is a layered approach, MILE entries can define overwrite conditions in order to express specific constraints set by the transfer context (e.g. 'target direct object must be in plural') without influencing the monolingual description of an entry.

4.2 XLIFF

This initiative, under the umbrella of the OASIS initiative (*www.oasis-open.org/committees/xliff*), deals with localisation aspects. If a translation job is handed to an agency, usually the text to be translated, the terminology to be used, and the translation memories to be consulted, are delivered in a package. The goal of XLIFF is to standardise the format of such a package: "*The purpose of the OASIS XLIFF TC is to define, through XML vocabularies, an extensible specification for the interchange of localisation information*" (XLIFF V1.1 White Paper 2003).

XLIFF focuses mainly on text handling and translation memory exchange; for terminology exchange, TBX is proposed as standard. There are no specific activities towards terminology exchange.

4.3 Ontology Languages

Ontology languages, the most notable of them being the Web Ontology Language OWL (*www.w3.org/TR/owl-features*), are used to describe meta-information in the context of the semantic web; they describe links between the nodes of the ontology, and rules for derivations and formal properties to be taken care of.

They do not describe any linguistic properties of the ontology concepts, and rarely worry about multilingual issues; the general assumption is that the ontology is a language-independent phenomenon, and each node of the ontology is represented by multilingual terms. This approach is rather similar to the TBX concept.

4.4 Lexical Markup Framework

Recent developments in the effort of standardisation have moved away from the straightforward DTD-based approaches into more general domains of standardisation frameworks, as the efforts for TMF (Terminological Markup framework), supposed to cover both the MATER and the GENETER exchange variants), or for LMF (Lexical Markup Framework) show. The basic idea is to separate two aspects of the exchange formats:

1. The basic data elements to be exchanged, i.e. the **data categories**. This effort, which covers many languages, provides e.g. attributes and values to describe gender, part-of-speech, and other linguistic information items to be exchanged.
2. The way how such data categories can be organised through the provision of **meta-models**; the idea is that implementations like TBX, OLIF and others are just instances of some more abstract meta-model which in turn can cover complete families of exchange formats.

Projects like LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems), an eContent project (*http://lirics.loria.fr/*), and efforts in the context of ISO (TC37/SC4, *http://www.tc37sc4.org/*) try to promote these approaches.

The efforts for standards on data categories could overcome the weakness of the current descriptions in TBX, OLIF, even EAGLES, namely that they support only some of the information categories, and only for some languages. Every time new languages, or new phenomena, are added, the standards need to be revised. A more systematic effort would help to achieve easier exchange of such data, and would also enable resource providers to define easy access using e.g. a common API for morphosyntactic access as defined in LIRICS.

As far as the definition of meta-models is concerned, the challenge is to find a balance between a very abstract model which covers any possible configuration, and proposals which can be implemented and used for exchange of concrete data. It could be proven that both TBX and OLIF can be described with the LMF meta-models, and so could possibly many others.

References

Atkins, S., Bel, N., et al. (2002). "From Resources to Applications. Designing the Multilingual ISLE Lexical Entry". In: Proceedings LREC III, Gran Canaria.

Calzolari, N., Grishman, R., Palmer, M. (2002). "Standards & Best Practice for Multilingual Computational Lexicons: ISLE, MILE and More". In: Proceedings LREC III, Gran Canaria.

Hegel, G. F. W. (1807). "Phänomenologie des Geistes".

Hohnhold, I. (1984). "The TEAM Terminology Data Bank System". In: TermNet News. Journal of the International Network for Terminology, pp. 19-33.

Hohnhold, I., Schneider Th. (1991). "Terminological Records and Lexicon Entries. A Contrastive Analysis". In: META 36, pp. 161-173.

Ide, N., Lenci, A., Calzolari, N. (2003). "RDF Instantiation of the ISLE/MILE Lexical Entries". In: Proceedings ACL Workshop on Linguistic Annotation: Getting the Model Right, Sapporo.

McCormick, S., Lieske C. (2005). "OLIF Tutorial". *http://www.berlinopenforum.de/download/christian_Lieske.zip*

Thurmair G., Lieske, C. (2002). "Lexical Exchange Formats – DXLT and OLIF". In: Proceedings of LRC Workshop, Dublin.

Vossen, P. (ed.) (1999). "EuroWordNet. General Document". EWN Project Report.

XLIFF V1.1 White Paper 2003. *http://www.oasis-open.org/apps/group_public/download.php/3110/XLIFF-core-whitepaper_1.1-cs.pdf*

(Endnotes)

[1] Only noun, verb, adjective and other are foreseen.

[2] It could be claimed that the example refers to different concepts. This is true and shows that most MT dictionaries are not concept-based. However, even within a concept there can be different translations; this was the starting point to develop language-specific concept hierarchies in EuroWordNet.

[3] The idea to characterise an entry by an ID is not sufficient in an exchange format, as both the dictionary where the entries come from and the dictionary where they go to have their own ID systems, and just using IDs in a foreign environment would not really help; an explicit meaning description is required.

[4] The definition of an entry on a semantic base raises a huge amount of challenges: How to define it, how to decide on one or several concepts, what about metaphors etc. This is a vast research area, which is explored in lexical semantics, WordNet or FrameNet. However, it is outside of the OLIF standard: Whatever is decided to be a concept can be exchanged in the OLIF format.

Monica Gavrila, Walther von Hahn, Cristina Vertan

ManageLex
A Tool for the Management of Complex Lexical Structures

Abstract

This paper describes MANAGELEX, a lexicon management tool, developed at Hamburg University, Natural Language Systems Division. After a general introduction on lexicons, the authors present the architecture and functionality of MANAGELEX. Sections 3 and 4 give information on two of the MANAGELEX modules concerning the choice and the structural organization of the linguistic features in a lexicon.

1 Introduction

In both monolingual and multilingual environments, language resources play a crucial role in preparing, processing and managing the lexical information and knowledge needed by computers. A large variety of computational lexicons was created, leading to a huge amount of different lexical structures and formats. This variety was triggered by differences between languages, differences in purpose and content, and differences in linguistic theory. In the past, numerous small and medium size lexicons were built in projects and became non-reusable later on because of their specific linguistic model or non-standard format.

In order to reduce the work that is done repeatedly in creating lexicons, standard formats and models were created including

- **standard lexicon formats** like EAGLES (*http://www.ilc.pi.cnr/*), MILE (for details see Calzolari et al. 2003), SALT, etc.
- **standard lexicon models** like GeneLex, Multilex, Parole/Simple (PAROLE/SIMPLE Report 1, PAROLE/SIMPLE Report 2) etc.

However, for many applications these standards are too complicated (because they try to model everything), and still contain gaps in modeling features of less spoken languages. Sometimes, for projects or evaluations (a series of) smaller lexicons with specific or even changing specifications are needed.

Another problem is the complicated manipulation of the existing lexicons as stand-alone components; either some of them have been produced with acquisition / save tools that may not be maintained any longer or do not have flexible export facilities, or they may contain procedural elements dependent on the host system.

Another problem is the operation of merging lexicons. Especially for less spoken languages, merging several small lexicons developed in different projects is an important step towards the achievement of a computational lexicographic resource for that language. The merging of lexicons is complicated by several factors:

- differences in format and encoding which frequently do not match,
- differences in linguistic categories,
- inconsistencies of values or different granularities.

2 MANAGELEX

General lexical management tools, which help the user to manipulate and validate lexicons, represent an alternative to standardization. Such a tool is MANAGELEX, currently under development at Hamburg University. This tool is not intended for replacing the present standards, but for managing the already existing lexicons (standard or non-standard).

MANAGELEX is “a generic lexicon management tool” (Vertan/von Hahn 2002) that permits the user to create, read, convert, and combine lexicons. MANAGELEX is also intended to enable the merging of lexicons that do not share common import and export formats. It also enhances the reusability of lexicons created in earlier projects by providing a tool that makes it possible to convert lexicographical data. Its design is format-, language- and platform-independent. Following functionalities are to be supported in MANAGELEX (the GUI is shown in Figure 1):

– reading and saving different encoding formats;
– accessing, creating and transforming different lexicon structures;
– merging of two lexicons either by merging their structure or merging lexical entries.

The main goals of MANAGELEX consist in improving the reusability of lexicons and and facilitating lexicon handling without dictates of a standard format or model.

2.1 Architecture

The MANAGELEX architecture follows the ANSI specification and contains three levels: real word v, model level and meta-model level.

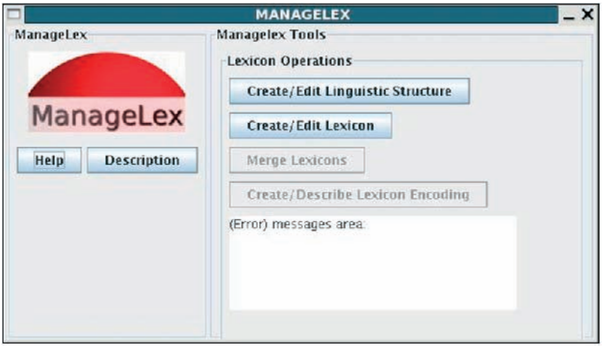


Fig. 1: MANAGELEX GUI Snapshot

Real, distinct objects represent the real world level, i.e. files that consist of the lexicon structure (Structure files: ***StructA***), files that contain the encoded lexicon (Lexicon files: ***DocA***), and lexicon content files (***EntryA***).

The model level consists of four tools:

1. **EditTool** reads, adds or updates entries in a lexicon.
2. **StructTool/LexTool** defines or updates the linguistic specification of a lexicon. This tool is described in detail in Section 4.
3. **EncodTool** decodes lexicon files and encodes lexicon entries into files. The encoding/decoding operation is done according to the specification in EncodMode, or, where it is missing, according to the user specification.
4. **MapTool** merges two lexicons with possibly different linguistic specifications.

The **meta-model level** is composed of three models:

1. **LexMode** is a rich model of possible lexical information. It is described in detail in Section 3.
2. **EncodMode** specifies the data structure of a specific entry and of a specific lexicon. The model is to be built after analyzing several existing models (e.g. OLIF, SALT, etc.).
3. **MapMode** specifies how two lexicons can be mapped. It has to take into consideration mutual gaps, complex categories, etc.

For the moment only StructTool and LexMode are fully implemented.

2.2 Functionality

Following operations can be performed within MANAGELEX: building, reading, and updating a lexicon, and merging two lexi-

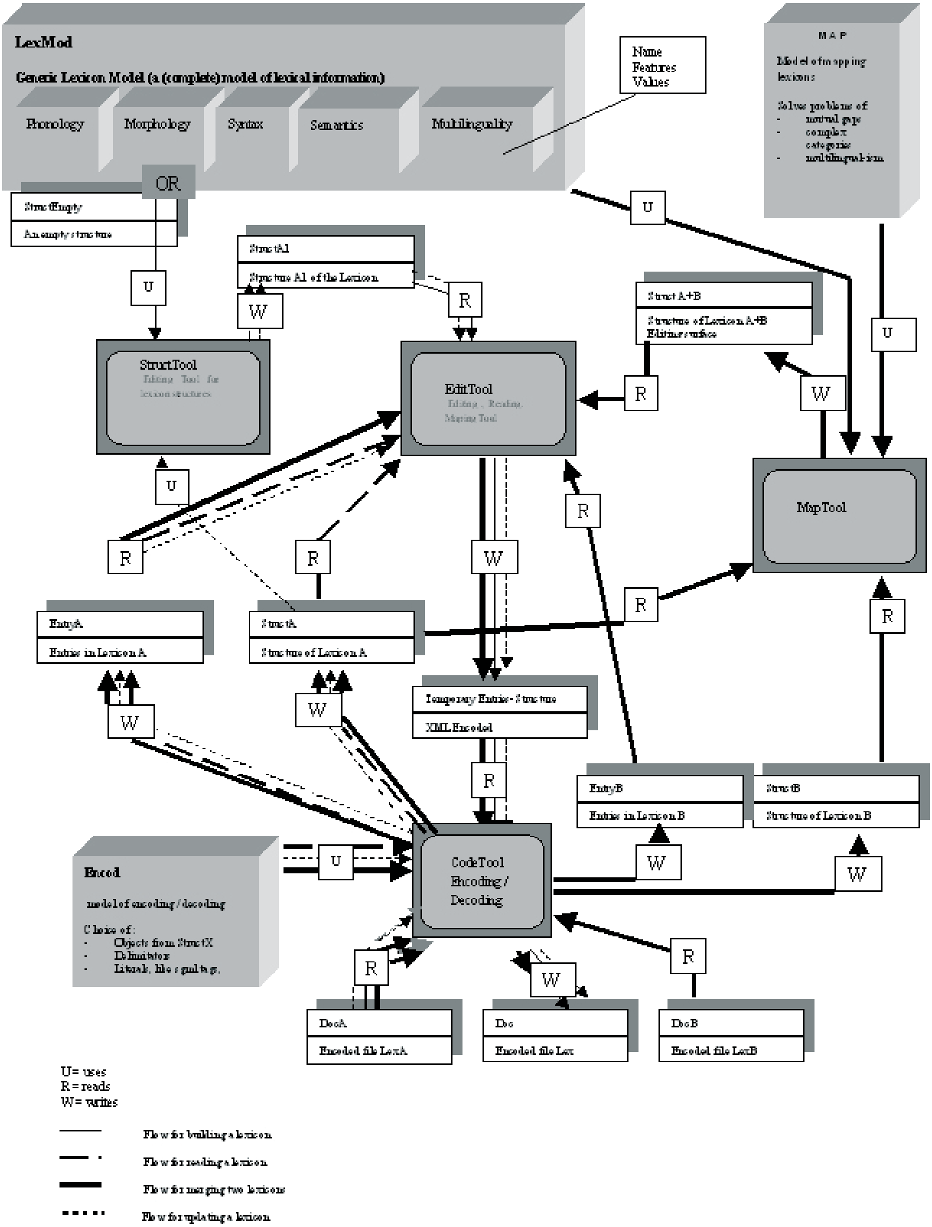


Fig. 2: MANAGELEX architecture and functionality (GAVRILA 2004)

cons. Figure 2 shows the architecture and functionality of MANAGELEX.

The most complex operation is the merge of two lexicons. It involves the use of LexMode, MapMode, EncodMode, and of three tools: the encoding/decoding tool (EncodTool), the mapping tool (MapTool), and the editing/reading tool (EditTool). Let us assume we want to merge Lexicons A and B. First, the encoding/decoding tool provides the two structure files StructA and StructB and two files containing the entries in the lexicon (LexA and LexB).

This operation is performed using EncodMode. The mapping tool uses the two structure files StructA and Struct B, LexMode and MapMode. As output is produced a new structure file which contains all linguistic elements of the two lexicons and in which all possible feature and value overlaps are resolved. The user will solve the overlapping problems if they cannot be resolved automatically. The mapping of the entries from the entry files and the new structure is done by the editing / reading tool. The sequence of these operations is illustrated in Figure 3.

3 LexMode– the Linguistic Resource in MANAGELEX

MANAGELEX is structured around three metamodels describing the linguistic information (LexMode), the encoding format (EncodMode) and the mapping between two lexicons (MapMode). In this section we will introduce LexMode – a generic ***lexicon model***, which aims to contain as much lexical information as possible. In this model, linguistic features and their possible values are specified. The model construction is based on the study of more than 12 machine-readable lexicons (e.g. CELEX, MULTEXT, GermaNet, Verbmobil) and of several standard lexicon models (e.g. PAROLE/SIMPLE and MILE). Most of the lexicon formats were analyzed in (Gius 2003). More details on lexicons can be found in (Handke 1995).

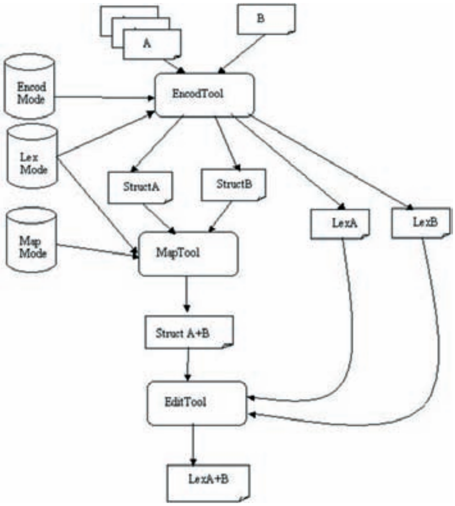


Fig. 3: Flow of operations for the merge of two lexicons in MANAGELEX

In the design of LexMode, we paid special attention to the separation between linguistic data and language data (e.g. examples can be added in the entries, but not in the lexicon structure).

LexMode can be updvated with new linguistic features specific to other languages or linguistic focus. Trying to be a lexicon model, it contains as much information as possible which also implies that no optional grammatical features are included. Because all features have cardinality constraints set on value one, it means that all information should be specified in a lexicon entry. In case this is not needed by the user, the cardinality constraints can be modified, or a new lexicon structure can be created. In order to preserve the generality LexMode also contains no relations between features (as in other lexicon models – e.g. PAROLE/SIMPLE – (Parole/Simple Report 1, Parole/Simple Report 2, Ruimy et al. 1998), but, if required, these can be specified later using StructTool.

The LexMode structure contains following levels of information:

- lexicon information,
- entry information,
- morphological information,
- phonological information,
- syntactical information,
- semantic information,
- multilingual information.

The structure of LexMode is presented in Figure 4.

3.1 Formal Specification of LexMode

In this section we will motivate our choice for the formal specification language used for LexMode, and present the way of describing LexMode in this language (OWL).

Three possibilities were considered for the language specification of LexMode: XML, RDF/RDFS, and OWL.

Due to the availability of manipulating tools, and transparency of the language, **XML** could have been a straightforward solution. A first drawback of this approach was the redundancies, which it can introduce. An example of such redundancy is shown below.

```
<category>
  <cname>Part of Speech</cname>
  <category>
    <cname>Noun</cname>
    <attribute>
      <aname>Gender</aname>
      <value>masculine</value>
      <value>feminine</value>
      <value>neuter</value>
    </attribute>
  </category>
  <category>
    <cname>Adjective</cname>
    <attribute>
      <aname>Gender</aname>
      <value>masculine</value>
    </attribute>
  </category>
</category>
```

We observe that the features and their values for nouns and adjectives, although quite similar, have to be repeated. This problem can be solved through the introduction of parameter variables in the DTD. However the new versions of meta-

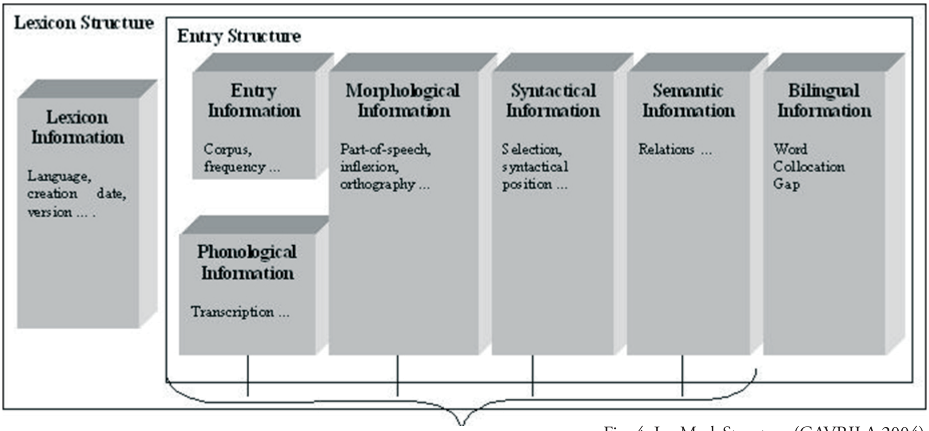


Fig. 4: LexModeStructure (GAVRILA 2004)

languages for XML (XML-Schema) do not preserve any longer this possibility, and its simulation through other mechanisms offered by the language is still cumbersome.

The second possibility – **RDF/RDFS** (Resource Description Framework Schema) allows data description by means of triples (Subject, Predicate, Object), and for an organization of information in classes and properties. However, it offers only a limited set of relations between classes and/or properties (e.g. class, subclassof, subpropertyof, but no synonymy) and it does not allow restrictions (e.g. cardinality restrictions).

The **Web Ontology Language OWL** (*http://www.w3.org/TR/owl-features*) which is currently based on RDF / XML was powerful enough for the description of LexMod. 15 out of the 48 existing tags were used: **owl:cardinality**, **owl:Class**, **owl:DatatypeProperty**, **owl: maxCardinality**, **owl: minCardinality**, **owl:ObjectProperty**, **owl:onProperty**, **owl:Restriction**, **owl: unionOf**, **rdf: List**, **rdf: type**, **rdfs:comment**, **rdfs:domain**, **rdfs:Literal**, and **rdfs:range**.

Another reason for choosing OWL was to facilitate the integration into the Semantic Web framework (Gavrila/Vertan 2005). Figure 5 gives a short example from the LexModeOWL file.

The example presents what parts of speech we chose in the morphological description.

In LexMode the distinction in describing grammatical features as class or property was done according to the following criteria:

- If a grammatical feature is described using other features, than it is a class.
- If there are relations between classes/features, then an object property is used.

```
<owl:ObjectProperty
     rdf:ID=”hasPartOfSpeech”>
<rdfs:domain
 rdf:resource=”#MorphologicalInfo”/>
<rdfs:range>
<owl:Class>
<owl:unionOf
 rdf:parseType=”Collection”>
<owl:Class rdf:about=”#Verb”/>
<owl:Class rdf:about=”#Noun”/>
<owl:Class rdf:about=”#Numeral”/>
<owl:Class rdf:about=”#Adjective”/>
<owl:Class rdf:about=”#Pronoun”/>
<owl:Class rdf:about=”#Determiner”/>
<owl:Class rdf:about=”#Article”/>
<owl:Class rdf:about=”#Conjunction”/>
<owl:Class rdf:about=”#Preposition”/>
<owl:Class rdf:about=”#VerbParticle”/>
<owl:Class rdf:about=”#Particle”/>
<owl:Class rdf:about=”#Adverb”/>
</owl:unionOf>
</owl:Class>
</rdfs:range>
</owl:ObjectProperty>
```

Fig. 5: Example from the LexModeOWL file

We also mark different literals and numbers when working with data type properties, as this is very useful for the merging operation.

The LexMode OWL encoding has 34 elements, 88 data type properties, and 23 object properties. In the table below we give some of the LexMode classes and properties.

The above organization of LexMode is flexible enough and fits into the MANAGE-LEX schema. This means that, apart from the role that LexMode is playing for the StructTool (starting point in creating a new structure), it also helps the MapTool in merging two lexicons structures.

The operations that can be done on linguistic categories are: adding, deleting, renaming, merging, and splitting. For example if one lexicon structure contains the verb category with the

Class	Property
LexiconStructure	hasLexiconInfo, hasEntryStructure
LexiconInfo	lexiconName, language, version, creationDate, modificationDate, copyright
EntryStructure	hasEntryInfo, hasMonolingualStructure, hasBilingualStructure
EntryInfo	corpus, frequency, workingState, termStatus, generationType, registeredEntry, refID, source
MonolingualStructure	hasMorphologicalInfo, hasPhonologicalInfo, hasSyntacticalInfo, hasSemanticInfo
BilingualStructure	toLanguage, toLexicon, hasCorrespondences
MorphologicalInfo	HasPOS, etc.
PhonologicalInfo	phoneticTranscription, terminalDevoicing, accents, audioFile
SyntacticalInfo	hasSelection, syntacticPosition, special
SemanticInfo	hasRelations, ontologyTypes, semanticFeatures, prototype, thematicRoles,

Table 1: Some classes and properties in LexMod

property transitivity and in the new structure there should be two different categories: transitive verb and intransitive verb, the category from the first structure is split into two different categories and the transitivity property is deleted.

4 Describing the Linguistic Structure

The structure tool (StructTool) allows the user to define the lexicon structure according to the particular application requirements. It allows the user to add, delete, merge, split, rename or select elements/grammatical features and create the needed lexicon structure (see Figure 6).

Following operations are implemented within StructTool:

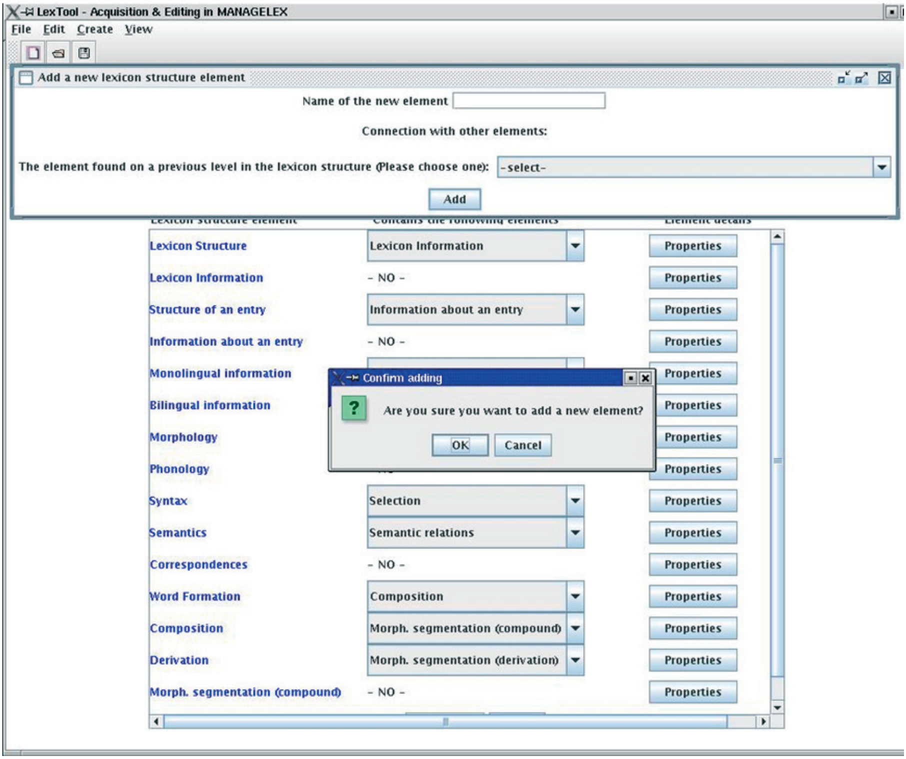
- reading LexMod,
- selecting categories and their values and/or ranges,
- defining new categories,
- updating values of existing categories,
- defining the structure of a lexicon,
- calling EditTool.

StructTool reads LexMode (or other OWL encoded lexicon structure files), generates a GUI that supports selections and editing (Add, Delete, Merge, Split, Rename operations on grammatical features) and saves a new StructX lexicon structure file.

As an example of an operation that can be performed with StructTool we present how a property is updated by renaming. If in a certain moment the user wants to rename an existing property, this can be easily done from the graphical interface. The process of renaming itself is a little bit more complicated, because the new name has to replace the old name in the whole lexicon structure – everywhere there is a reference to it -, so that lexicon structure consistency is kept.

5 Conclusions and Further Work

In this paper we presented MANAGELEX, a generic lexicon management tool that can be regar-


LexTool - Acquisition & Editing in MANAGELEX
File Edit Create View
Add a new lexicon structure element
Add
Confirm adding
OK
Cancel

Name of the new element

Connection with other elements:

The element found on a previous level in the lexicon structure (Please choose one): -select-

Are you sure you want to add a new element?

Lexicon structure element	Contains the following elements	Element details
Lexicon Structure	Lexicon Information	Properties
Lexicon Information	- NO -	Properties
Structure of an entry	Information about an entry	Properties
Information about an entry	- NO -	Properties
Monolingual information		Properties
Bilingual information		Properties
Morphology		Properties
Phonology		Properties
Syntax	Selection	Properties
Semantics	Semantic relations	Properties
Correspondences	- NO -	Properties
Word Formation	Composition	Properties
Composition	Morph. segmentation (compound)	Properties
Derivation	Morph. segmentation (derivation)	Properties
Morph. segmentation (compound)	- NO -	Properties

Fig. 6: StructTool GUI

ded as a possible alternative to lexicon standardization. The tool is flexible and easy to use also for non-specialists. For the moment, only European languages are modeled.

We are currently working at the implementation of the other tools and models of the MANAGELEX system as well as preparing ready-to-start configurations for widely used standards like PAROLE/SIMPLE and MILE. We are also planning an updating process for LexMod, by including important changes (adding operation) in structure files into LexMode. Further on, we would like to extend our linguistic model to other types of languages. Information regarding the last version of the system can be found at *http://nats-www.informatik.uni-hamburg.de/view/MainManageLex*.

References

Calzolari, N., Bertagna, F., Lenci , A., Monachini, M. (2003). “Standards and Best Practice for Multilingual Computational Lexicons &MILE (the Multilingual ISLE Lexical entry)”. Deliverable D2.2-D3.2 ISLE Computational Lexicon Working Group, *http://www.ilc.cnr.it/EAGLES96/isle/clwg_doc/ISLE_D2.1-D3.1.zip*.

Gavrila, M. (2004). “LexMod and LexTool – Lexical Model, Acquisition and Editing in MANAGELEX”, Diploma Thesis, Hamburg University, Computer Science Faculty, R36887.

Gavrila, M., Vertan C. (2005).“MANAGELEX and the Semantic Web”. OntoLex Workshop Proceedings, IJCNLP-05, Jeju, South Korea, October 2005.

Gius, E. (2003). “Vergleich maschinenlesbarer deutscher Lexika nach linguistischem Inhalt, Wertebereichen und Kodierung”, Diploma thesis, University of Hamburg, manuscript.

Parole/Simple Report 1. “Report on the Syntactic Layer”, *http://www.ub.es/gilcub/SIMPLE/reports/parole/parole_syn/parosyn.html* .

Parole/Simple Report 2. “Report on the Morphological Layer”, *http://www.ub.es/gilcub/SIMPLE/reports/parole/parole_morph/paromor.html* .

Handke, J. (1995). “The Structure of the Lexicon – Human versus Machine”. Berlin-NewYork: Mouton de Gruyter, .

Ruimy, N., Corrazzari, O., Gola, E., Spanu, A., Calzolari, N., Zampolli, A. (1998). “The European LE-PAROLE Project and the Italian Lexical Instantiation”. In: Proceedings of the ALLC/ACH, 1998, Lajos Kossuth University, Debrecen, Hungary, July 1998, pp. 149-153:

Vertan, C., von Hahn, W. (2002). “Towards a Generic Architecture for Lexicon Management”. In: Proc. LREC-2002. Third International Conference on Language Resources and Evaluation. Las Palmas de Gran Canaría, May 2002.

von Hahn, W. (2005). “Merging Computer-Readable Heterogeneous Terminological Material”. In: Proceedings of the XVth European Symposium of Languages for Special Purposes (LSP’05), Bergamo.

Georg Heeg

Flexible Technologies to Visualize and Transform Terminological Representations

Modelling Representations instead of Programming using Smalltalk

Abstract

This paper discusses a software design approach to allow interchange of linguistic data. It focuses on the modelling of the linguistic concepts represented in the data and describes the transfer between exchange formats as a multi-tier interpretation/generation. These concepts are implemented in Smalltalk, a programming environment enabling flexible conversion of data between formats supported by Terminology Management Systems (TMS).

1 What is the Issue?

Most of today's software suffers from being inadequate in its innermost part. It is constructed from bits and algorithms.

Many papers in the workshop which have been documented in this issue of LDV Forum describe the tedious tasks to transfer contents from one terminology system to another. This issue is perceived as difficult because the domain of transfer requires an understanding of different domains at the same time.

This paper describes the use of Smalltalk to understand these different domains and to provide an implementation of the transfer problem at the same time. For Smalltalk, the key task is modelling instead of programming, and thus Smalltalk closes the gap between human thinking and its implementation in software.

The main question shifts from "How shall a particular feature be executed?" to "Who is responsible for a particular task?"

2 Modelling vs. Programming

2.1 Modelling in the Good Old Days

Before computers were invented in the mid 20th century, all computing was done by people, all algorithms were executed manually and it was important to represent knowledge in terms understandable to humans.

Where formalisms were needed, forms had been developed which resembled the thinking and terminology of the domain in question.

To give an example, assume you had a toothache in those days before computers were installed in dental practices. So you walked to your favourite dentist. She looked at you from her professional point of view and she recognized you as her patient. Additionally, she had a form which contained all her terminology produced by professional dentist publishers whose primary goal was "ease of use". This was realized by a good mapping of dentists' concepts on paper (see Fig. 1).

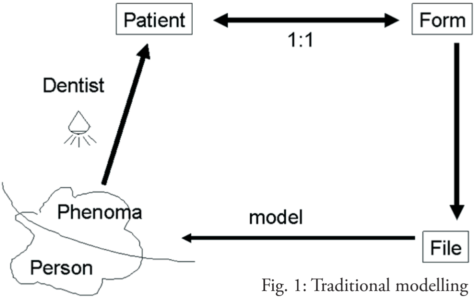


Fig. 1: Traditional modelling

The file which is essentially the filled form represented the patient's phenomena of interest to the dentist in an adequate and understandable form.

2.2 Modelling in the Computer Days

After computers were invented in the 1940s they were extremely expensive. In the 1970s, mainframe computers were 100 times slower and larger than today's PCs. One hour of usage of such a mainframe amounted to one monthly salary for a programmer. In this spirit most of computer technology was developed. The common mind set was and is that computers are expensive and thus it is mostly important to represent all information and procedures the computer way. And computers have two major components: CPU and Memory.

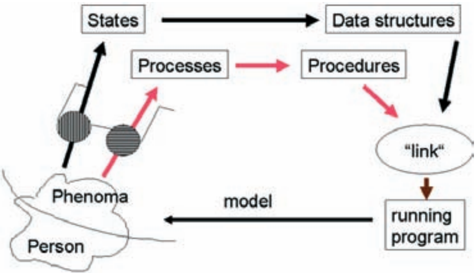


Fig. 2: Traditional computer modelling

This mind set influences still today many software projects and starts from the analysis phase. Figure 2 illustrates the traditional computer modelling process. From the very beginning the question is different: Instead of looking at the ontology of a domain, the viewpoint uses a filter to search for data and procedures. The following examples illustrate the limitations of this approach:

Example 1

1. Wooden body in the form of a cylinder with approx. 20 cm (8 inch) height and 6 mm, (1/4 inch) in diameter.
2. The centre of the cylinder contains a drilling of 1 mm filled with pressed graphite.
3. At one end, the cylinder is conically tapered.
4. The graphite can be transferred to other bodies by rubbing.

Example 2

1. Plastic tube in the form of a cylinder with approx. 20 cm (8 inch) height and 6 mm (1/4 inch) in diameter.
2. Inside is another plastic tube with 2 mm (1/12 inch) in diameter and at the top there is a metal ball.
3. The inner tube is filled with a viscous liquid.
4. The liquid can be transferred to other media with the help of the ball.

In both examples the first three issues describe state/information while the last one describes process/procedural aspects.

Common Sense

Reading above descriptions a normal (non-computer) person will easily recognize that the first is a strange description of a pencil while the latter is a not less strange description of a ball pen.

When I ask a normal person (or even better a child) they will come up with a totally different description of pencil and ball pen: They serve to draw and write and the main difference is that using a pencil you can easily use an eraser to rub out.

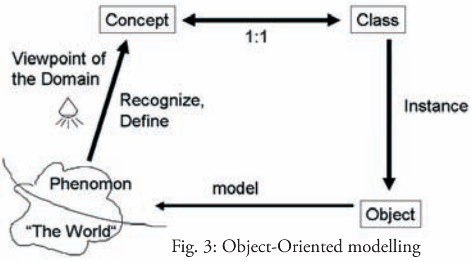


Fig. 3: Object-Oriented modelling

2.3 Modelling like in the Old Days

The basic idea of the programming language Smalltalk is to go back to this "naïve" understanding and to map concepts directly into software (see Fig. 3). The key idea resembles the strategy in the pre-computer times: understandability and adequacy.

2.4 Objects and Data

Objects in object-oriented languages like Smalltalk primarily care about the services they provide. Data are encapsulated inside the objects and are not visible from the outside.

3 Applying Objects to Transfer Problems

Now let us start to apply this approach to the exchange of lexical and terminological resources.

To do so, we first analyse the problems in this area and categorize them into different kinds of problems.

Similarly to communication levels in general, one can observe in our domain three different types of problems: Syntactic, structural and content mismatches.

3.1 Syntactic Mismatch

Some systems produce exchange files in plain text files, one line per entry, separated by special delimiters like "@@@". If this delimiter is a comma, these text files are often called CSV files (comma separated values); if it is a tabulator character these files are called TSV files (tab separated values). As these two file types can be read with Microsoft Excel, they are often called Excel files.

Another file format, which has become very common recently, is XML (eXtensible Mark-up Language). XML files are annotated trees represented in a linear fashion. Obviously, this approach offers more flexibility than plain text or Excel files, as entries with differently structured information can be stored in the same file.

3.2 Structural Mismatch

Let us assume we have a match in the syntactic form of a file, we can still have strong incompatibilities. The structures of the files do not match.

Examples are: Excel file columns do not match; XML files have different DTDs (DTDs and schemata describe the structure of XML files).

3.3 Semantic Mismatch

File contents do not necessarily match even if all structures match. There is still room for incompatibilities. Examples are: Some systems allow importing language pairs only; others allow importing many languages at the same time. Some allow multiple entries to represent homonymic terms, other require special entries.

4 SSS-TTT

As we found three levels of problems we will also start with three levels (also called tiers) of solutions: our architecture implements a syntactic, structural, semantic three-tier transfer (SSS-TTT).

Additionally, we have the desire to get a universal converter which can convert any input form to any output form. Universal converters are much easier to design in a so-called star architecture which consists of a common internal representation which can be filled by any input converter and can output to any destination format.

In this particular case of a layered problem description universal converters can be developed for each tier reducing the number of converters even further providing standardized interfaces between the tiers.

The syntactic tier converter reads the files and maps keys to values. Keys can be column numbers, column headers or XML entity labels, values normally are strings, for inner XML nodes values are trees.

Converters of the structural tier map logical attributes onto (potentially multiple keyed) values.

These values are transformed in the semantic tier into "meaningful" objects.

If needed, transformation and filtering is done on the level of meaningful objects, the result of the semantic layer.

For each converter there is a generator which operates in the opposite direction.

4.1 Implementation in Smalltalk

For the syntactic tier Smalltalk provides predefined classes. in particular an XML Parser/Generator and a CSV/TVS Reader/Writer.

Additionally, there is a bunch of technologies available to communicate directly to language software. COM Connect and .NET Connect use Microsoft inter-process communication, WebServices and Corba over IIOP connect into the Java world.

In Smalltalk, everything is in source and everything can be enhanced, it can be changed and adapted whenever needed.

The structural and semantic tiers map linguistic theories easily into software artefacts.

4.2 Smalltalk Development Process

Smalltalk always gives you immediate feedback, thus it fully supports agile programming. Thus it is good practice to interleave programming and testing all the time: "Make it work half way", "Try it out", "Make it work a little bit more", "Try it out again".

A well known development strategy in Smalltalk is:

1. Make it work
2. Make it right
3. Make it fast (if needed)

The main technique in steps 2 and 3 is called refactoring. "Refactoring is the process of rewriting a computer program or other written material to improve its readability or structure with the explicit purpose of keeping its meaning or behaviour" (Wikipedia 2006).

5 Linguistic Smalltalk Experiences

In cooperation between the Software Localization Group of Anhalt University of Applied Sciences and Georg Heeg eK (*http://www.heeg.de/*) several Smalltalk projects have been successfully developed. They fall into three categories: Small transfer tools to get data into professional MT, CAT and TMS systems, tools for Software Localization education and Software Localization research. One of these projects, which consisted of making Microsoft glossaries accessible for Software tools, demonstrates typical problems and their solutions. Therefore, we want to describe this project in more detail.

Microsoft provides its products in many languages. As described on the web page *http://www.lai.com/microsft.html* Microsoft provides its translation catalogs in 24 languages on ftp server *ftp://ftp.microsoft. com/developr/msdn/newup/glossary*.

When you unzip all files you will get 5.8 GB *.csv files. Most of them are too big to open them in Microsoft Excel. So other tools are needed to get access to this very rich resource.

When you try to import these glossaries you will see additional problems: From language to language the number of columns differs; several files contain no headers at all.

Mostly the filename of the glossary files indicates the language and country of the translations, but some language codes are represented with 3 characters, as German in "deu-deu-Access2003.csv", others with 2 characters, as Czech in "cz_vb50.csv", which are different ISO standards.

All of these problems have two things in common: There is no description at all and you step over them just by accident. So it is excellent to have an open flexible tool like Smalltalk with full control for the developer.

To read the files we started with subclassing CSVReader. In some of the CSV files the entries are separated by commas, in others by tabulators. We made CSV Reader doing the right guess.

Then we saw that the number of columns ranges from 8 to 255; we looked at the data and guessed the intention.

Some of the files had no columns headers at all, so we added guessing the column structure in CSVReader subclass.

Some of the files have a comment in the first line; our guess was easy: if the number of columns is 1, it is a comment.

As already mentioned, languages are indicated in filenames using different standards for the language codes, like in "deu-deu-Access2003.csv" or "cz_vb50.csv", so that the codes had to be transferred to homogeneous representations.

All files for the largest language pair (English -German) could be made available in a Smalltalk system, but all files for all languages (5.8 GB) cannot be loaded into current 32 bit VisualWorks systems. This requires a 64 bit version or an object database.

After reading you have a collection of objects representing the contents of all files read. These objects can be sorted by any sorting criteria, filtered anyhow, or matched against any input in a translation memory manner.

Last but not least these objects can be exported to any desired format. This allows loading subsets of the Microsoft glossaries into any translation memory or terminology management system. Examples are TMX and TBX files.

Certainly, the glossary tools can also be used in VisualWorks language tools developed at Georg Heeg eK and Anhalt University like LioN (Localizer for VisualWorks applications, see Lannatewitz 2003 and Haase 2005) and Web-TCM (Localizer for HTML-Pages; see Seewald-Heeg 2001).

6 Conclusion

Smalltalk serves as an ideal technology for linguistic tasks. It enables to create transformations between different terminological representations and modify them to get ad-hoc problems solved instantaneously. It is easy to try out new ideas and to observe the execution in graphic user interfaces immediately.

The main thing is "modelling instead of programming" to keep the entire software totally understandable.

References

Haase, C. (2005). "Lokalisierung einer Entwicklungsumgebung mit einem Nicht-Standard-System am Beispiel von LioN". Diploma Thesis, Anhalt University of Applied Sciences. *http://www.heeg.de/downloads/vortraege/DiplomArbeit-ClaudiaHaase-2005.pdf.*

Lannatewitz, D. (2003). " Entwicklung und Implementierung eines Lokalisierungswerkzeuges für VisualWorks". Diploma Thesis, Anhalt University of Applied Sciences, manuscript.

Seewald-Heeg, U. (2001). "Entwicklung und Einsatz von Lokalisierungswerkzeugen. Informatik-, Computerlinguistik-, Fachsprachenkompetenz". *http://www.heeg.de/~uta/PPT/Web-TCM/LokalisierungmitWeb-TCM.ppt.*

Wikipedia:Refactoring (2006). Refactoring – Wikipedia, The Free Encyclopedia, *http://en.wikipedia.org/wiki/Refactoring, accessed May 2006.*

Rachel Herwartz und Birgit Wöllbrink

www.terminologieforum.de
The Internet Discussion Platform for Terminological Subjects

Abstract

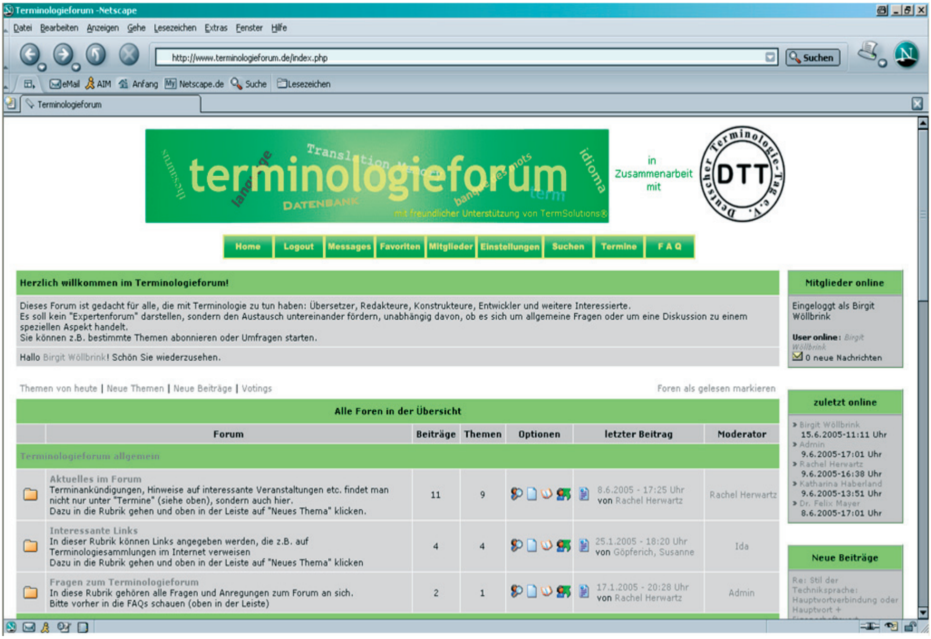
The internet discussion platform *www.terminologieforum.de* intends to be an independent, central and non-commercial meeting point for terminologists, translators, and technical writers to exchange knowledge on terminological subjects.

1 Idea

After having founded a consultancy in the fields of terminology and translation management and solutions) in 2004, Rachel Herwartz realized that there was no central and independent meeting point on the internet to discuss terminological subjects.

On the one hand, there are web discussion platforms which also cover issues in the field of terminology work and translation such as e.g. the “tekom webforum” (*www.tekom.de*) or the BDÜ discussion platform. However, these are not independent as they are only accessible to members of the respective associations. Their intention is to support expert discussions among the associations’ members.

On the other hand, there are web portals such as “DTP – Deutsches Terminologie-Portal”


Terminologieforum -Netscape
Datei Bearbeiten Anzeigen Gehe Lesezeichen Extras Fenster Hilfe
http://www.terminologieforum.de/index.php
Suchen
eMail AIM Anfang Netscape.de Suche Lesezeichen
Terminologieforum
terminologieforum
Translation
DATENBANK
term
idioma
mit freundlicher Unterstützung von TermSolutions®
in Zusammenarbeit mit
DTT
Deutscher Terminologie-Tag e.V.
Home
Logout
Messages
Favoriten
Mitglieder
Einstellungen
Suchen
Termine
F A Q

Herzlich willkommen im Terminologieforum!

Dieses Forum ist gedacht für alle, die mit Terminologie zu tun haben: Übersetzer, Redakteure, Konstrukteure, Entwickler und weitere Interessierte.
Es soll kein "Expertenforum" darstellen, sondern den Austausch untereinander fördern, unabhängig davon, ob es sich um allgemeine Fragen oder um eine Diskussion zu einem speziellen Aspekt handelt.
Sie können z.B. bestimmte Themen abonnieren oder Umfragen starten.

Hallo Birgit Wöllbrink! Schön Sie wiederzusehen.

Themen von heute | Neue Themen | Neue Beiträge | Votings   Foren als gelesen markieren

Alle Foren in der Übersicht

	Forum	Beiträge	Themen	Optionen	letzter Beitrag	Moderator
	Terminologieforum allgemein					
	Aktuelles im Forum Terminankündigungen, Hinweise auf interessante Veranstaltungen etc. findet man nicht nur unter "Termine" (siehe oben), sondern auch hier. Dazu in die Rubrik gehen und oben in der Leiste auf "Neues Thema" klicken.	11	9		8.6.2005 - 17:25 Uhr von Rachel Herwartz	Rachel Herwartz
	Interessante Links In dieser Rubrik können Links angegeben werden, die z.B. auf Terminologiesammlungen im Internet verweisen Dazu in die Rubrik gehen und oben in der Leiste auf "Neues Thema" klicken	4	4		25.1.2005 - 18:20 Uhr von Göpferich, Susanne	Ida
	Fragen zum Terminologieforum In diese Rubrik gehören alle Fragen und Anregungen zum Forum an sich. Bitte vorher in die FAQs schauen (oben in der Leiste)	2	1		17.1.2005 - 20:28 Uhr von Rachel Herwartz	Admin

Mitglieder online

Eingeloggt als Birgit Wöllbrink

User online: Birgit Wöllbrink
0 neue Nachrichten

zuletzt online

- Birgit Wöllbrink 15.6.2005-11:11 Uhr
- Admin 9.6.2005-17:01 Uhr
- Rachel Herwartz 9.6.2005-16:38 Uhr
- Katharina Haberland 9.6.2005-13:51 Uhr
- Dr. Felix Mayer 8.6.2005-17:01 Uhr

Neue Beiträge

Re: Stil der Techniksprache: Hauptwortverbindung oder Hauptwort +

Fig. 1: Introduction, navigation bar etc.

(*www.iim.fh-koeln.de/dtt/*), the web portals of the University of Innsbruck (Austria), Saarland University and University of Leipzig (Germany), Vaasa University (Finland), or "ETIS – European Terminology Information Server", see also (Spaetling 2002). These portals only provide information (in different forms and with different focus on terminology issues), but do not offer a possibility for discussion.

Keeping this in mind, the idea of an open, independent and central discussion platform on terminological subjects was born.

2 Status, Members and Associates

www.terminologieforum.de started in January 2005 and is an individual website, supported by TermSolutions® (*www.term-solutions.com/*) in providing webspace and technical support.

It is a non-commercial, non-profit website, open and free for any interested person.

Up to now, the forum has approx. 50 registered members, mostly technical writers and translators.

The forum cooperates with the DTT (Deutscher Terminologie-Tag e.V.) and its various boards are moderated by e.g. Prof. Dr. Klaus-Dirk Schmitz (FH Cologne) or Dr. Felix Mayer (SDI Munich).

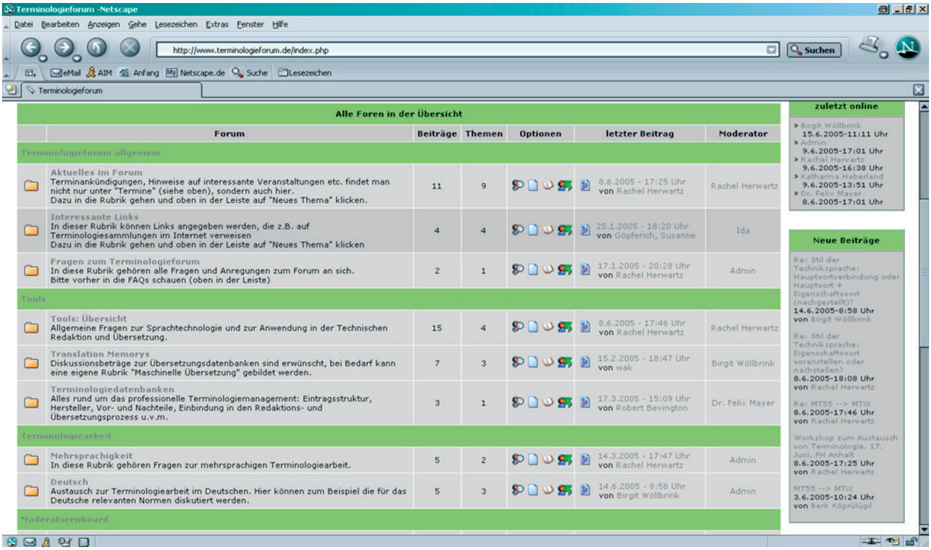
3 Structure

Figure 1 shows the start page and navigation bar of the "terminologieforum", where you can find a "Search" function, a list of members, a calendar of events, a list of links.

On the right hand side, information on the latest contributions and their status as well as on the members last online is given.

Below this "introductory area" you will find the "discussion area", containing several categories and their boards to post contributions (see Fig. 2).

Current user languages are German and English, further languages will follow.


Terminologieforum -Netscape
Datei Bearbeiten Anzeigen Gehe Lesezeichen Extras Fenster Hilfe
http://www.terminologieforum.de/index.php
Suchen
eMail AIM Anfang Netscape.de Suche Lesezeichen
Terminologieforum

Alle Foren in der Übersicht

Forum	Beiträge	Themen	Optionen	letzter Beitrag	Moderator
Terminologieforum allgemein					
Aktuelles im Forum Terminankündigungen, Hinweise auf interessante Veranstaltungen etc. findet man nicht nur unter "Termine" (siehe oben), sondern auch hier. Dazu in die Rubrik gehen und oben in der Leiste auf "Neues Thema" klicken.	11	9		8.6.2005 - 17:25 Uhr von Rachel Herwartz	Rachel Herwartz
Interessante Links In dieser Rubrik können Links angegeben werden, die z.B. auf Terminologiesammlungen im Internet verweisen Dazu in die Rubrik gehen und oben in der Leiste auf "Neues Thema" klicken	4	4		25.1.2005 - 18:20 Uhr von Göpferich, Susanne	Ida
Fragen zum Terminologieforum In diese Rubrik gehören alle Fragen und Anregungen zum Forum an sich. Bitte vorher in die FAQs schauen (oben in der Leiste)	2	1		17.1.2005 - 20:28 Uhr von Rachel Herwartz	Admin
Tools					
Tools: Übersicht Allgemeine Fragen zur Sprachtechnologie und zur Anwendung in der Technischen Redaktion und Übersetzung.	15	4		8.6.2005 - 17:46 Uhr von Rachel Herwartz	Rachel Herwartz
Translation Memorys Diskussionsbeträge zur Übersetzungsdatenbanken sind erwünscht, bei Bedarf kann eine eigene Rubrik "Maschinelle Übersetzung" gebildet werden.	7	3		15.2.2005 - 18:47 Uhr von wak	Birgit Wöllbrink
Terminologiedatenbanken Alles rund um das professionelle Terminologiemanagement: Eintragsstruktur, Hersteller, Vor- und Nachteile, Einbindung in den Redaktions- und Übersetzungsprozess u.v.m.	3	1		17.3.2005 - 15:09 Uhr von Robert Bevington	Dr. Felix Mayer
Terminologiearbeit					
Mehrsprachigkeit In diese Rubrik gehören Fragen zur mehrsprachigen Terminologiearbeit.	5	2		14.3.2005 - 17:47 Uhr von Rachel Herwartz	Admin
Deutsch Austausch zur Terminologiearbeit im Deutschen. Hier können zum Beispiel die für das Deutsche relevanten Normen diskutiert werden.	5	3		14.6.2005 - 8:58 Uhr von Birgit Wöllbrink	Admin
Moderatorenboard					

zuletzt online

- Birgit Wöllbrink
  15.6.2005-11:11 Uhr
- Admin
  9.6.2005-17:01 Uhr
- Rachel Hervartz
  9.6.2005-16:38 Uhr
- Katharina Haberland
  9.6.2005-13:51 Uhr
- Dr. Felix Mayer
  8.6.2005-17:01 Uhr

Neue Beiträge

Re: Stil der Techniksprache: Hauptwortverbindung oder Hauptwort + Eigenschaftswort (nachgestellt)?
14.6.2005-8:58 Uhr
von Birgit Wöllbrink

Re: Stil der Techniksprache: Eigenschaftswort voranstellen oder nachstellen?
8.6.2005-18:08 Uhr
von Rachel Hervartz

Re: MT55 --> MTiX
8.6.2005-17:46 Uhr
von Rachel Hervartz

Workshop zum Austausch von Terminologie, 17. Juni, FH Anhalt
8.6.2005-17:25 Uhr
von Rachel Hervartz

MT55 --> MTiX
3.6.2005-10:24 Uhr
von Berk Köprülügil

Fig. 2: Categories and boards of *terminologieforum.de*

The forum is subdivided into the following categories in which members can enter and read contributions to the respective subjects:

- terminology work (including the boards German, Multilingual),
- tools,
- terminology databases,
- translation memories,
- software localization.

Furthermore, the forum provides a collection of links and a calendar of events.

Following questions have been of special interest over the last few months:

1. Which literature can be recommended for students in the field of terminology work?
2. Where can you find information on the differences among the various translation memory and terminology tools or information on termcheckers?

Further issues have been the provision of rules on how to form denominations as well as questions on converting data from one terminology database into another.

4 Prospects

The discussions are currently followed by many interested persons so that the forum provides a productive exchange of information.

If desired by the members, new categories (e.g. "Machine Translation") can be introduced at any time. A category to gather new subjects for master theses/Diplomarbeiten is planned as well.

References

Spaetling, G. A. (2002). "DTP – Deutsches Terminologie-Portal". In: Mayer, Felix et al. (eds.) (2002). DTT Symposion eTerminology. Akten des Symposions Köln, 12.-13. April 2002, Köln: Deutscher Terminologie Tag e.V., pp. 247-252.

Stefanie Geldbach, Uta Seewald-Heeg

Appendix

The efforts toward standardization have sparked off numerous research projects and initiatives which have proposed and developed various standardized formats or lexicon models. While some of these standards (e.g. TMX) already have found wide acceptance and are supported by a growing number of commercial applications the fate of others is still undecided. Only future developments will show whether a given standard will actually be adopted by the language industry or rather be replaced by formats yet to come. At any rate, standardization of terminological and lexical resources is an important research field which will doubtlessly receive considerable attention also in the years to come. In order to facilitate the orientation within this dynamic field we decided to compile a short reference section which contains a – surely not exhaustive – list of past and ongoing standardization projects, file formats and organizations actively engaged in standardization issues. This list contains short definitions of important terms and projects discussed in this issue and provides links to websites for further reference.

1 Projects

ISLE (International Standards for Language Engineering). International project which aimed at improving the accessibility and availability of language resources. See *www.ilc.cnr.it/EAGLES96/isle/ISLE_Home_Page.htm*.

LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems). This ongoing project aims at providing ISO ratified standards for language technology to enable the exchange and reuse of multilingual language resources. See *http://lirics.loria.fr/*.

MILE (Multilingual ISLE Lexical Entry). Model for the lexical representation which is based on an open core set of data categories. See for ex. *http://www.w3.org/2001/sw/BestPractices/WNET/ISLE_D2.2-D3.2.pdf*.

SALT (Standards-based Access to Lexicographical and Terminological Multilingual Resources). Open-source project for creating open standards in order to facilitate the integration of terminological and lexicographical resources. See *http://www.loria.fr/projets/SALT/*.

2 Formats

MARTIF (Machine Readable Terminology Interchange Format). SGML-based format for exchanging terminological data between and among different terminology database systems published by ISO TC37/SC3 in 1999. See *http:// www.iso.org/*.

TBX (Term Base eXchange). Open XML-based standard developed by the Localization Industry Standards Association (LISA) to support the exchange of terminological data, see *www.lisa.org/standards/tbx*.

TMX (Translation Memory eXchange). Open XML-based standard exchange format for translation memories developed by LISA to allow easier exchange of translation memory data between tools and/or translation vendors, see *www.lisa.org/standards/tbx*.

OLIF2 (Open Lexicon Interchange Format). Open XML-based format developed by the OLIF Consortium to support the exchange of terminology and MT lexicons, see *www.olif.net*.

XLIFF (XML Localization Interchange File Format). Specifically designed to support the localization of data. XLIFF has features for updating strings, revision control, marking different phases of the localization process, word count calculations, the provision of alternative or suggested language translations. XLIFF is an open standard. See *http://www.oasis-open.org/committees/xliff.*

3 Organizations

LISA (The Localization Industry Standards Association). Since 1990, the LISA Forums and Global Strategies Summits have been dedicated to delivering best practices and standards for facilitating international business. See *http://www.lisa.org/*.

ELRA (European Language Resources Association). Established as a non-profit organisation in 1995, ELRA's goal is to make available the language resources for language engineering and to evaluate language engineering technologies. See *http://www.elra.info/*.

ISO (International Organization for Standardization). Founded in 1947 as the successor of the National Standardizing Associations (ISA) established in 1926, ISO is a network of the national standards institutes of 156 countries. The organization develops International Standards required by business, government and society. See *http://www.iso.org/*.

OASIS (Organization for the Advancement of Structured Information Standards). Founded in 1993, OASIS is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. The OASIS Localization Technical Committees (TC) include the Translation Web Services TC and the XLIFF TC. See *http://www.oasis-open.org/*.

OSCAR (Open Standards for Container/Content Allowing Re-use). OSCAR is LISA's working group for the development and maintenance of open standards for the language industry. Founded in 1997, OSCAR is dedicated to the development of open standards. Standards worked out by OSCAR include: Translation Memory eXchange (TMX) – the certifiable standard for the exchange of translation memory data; – the standard for exchange of structured terminological data; and Segmentation Rules eXchange (SRX) – the standard for exchange of information about how translation tools segment text. See *http://www.lisa.org/sigs/oscar/*.

Autorenverzeichnis

Monica Gavrila
Universität Hamburg
Fakultät für Mathematik, Informatik
und Naturwissenschaften
Department Informatik
Natural Language Systems
Vogt-Koelln-Str. 30
D-22527 Hamburg
gavrila@nats.informatik.uni-hamburg.de

Stefanie Geldbach
Rombachweg 11a
D-69118 Heidelberg
stefanie_geldbach@yahoo.de

Walther von Hahn
Universität Hamburg
Fakultät für Mathematik, Informatik
und Naturwissenschaften
Department Informatik
Natural Language Systems
Vogt-Koelln-Str. 30
D-22527 Hamburg
vhahn@informatik.uni-hamburg.de

Georg Heeg
Georg Heeg eK
Mühlenstr. 19
D-06366 Köthen
georg@heeg.de

Rachel Herwartz
TermSolutions
Mühlstraße 10
D-88085 Langenargen
info@term-solutions.com

Uta Seewald-Heeg
Hochschule Anhalt (FH)
Fachbereich Informatik
Computerlinguistik und Fachübersetzen
Lohmannstraße 23
D-06366 Köthen
uta.seewald-heeg@inf.hs-anhalt.de

Gregor Thurmair
linguatec Sprachtechnologien GmbH
Gottfried-Keller-Straße 12
D-81245 München
g.thurmair@linguatec.de

Cristina Vertan
Universität Hamburg
Fakultät für Mathematik, Informatik
und Naturwissenschaften
Department Informatik
Natural Language Systems
Vogt-Koelln-Str. 30
D-22527 Hamburg
vertan@informatik.uni-hamburg.de

Birgit Wöllbrink
TermSolutions
Mühlstraße 10
88085 Langenargen
woellbrink@term-solutions.com

Wolfgang Zenk
Acolada GmbH
Liliencronstr. 13
D-90429 Nürnberg
w.zenk@acolada.de