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.

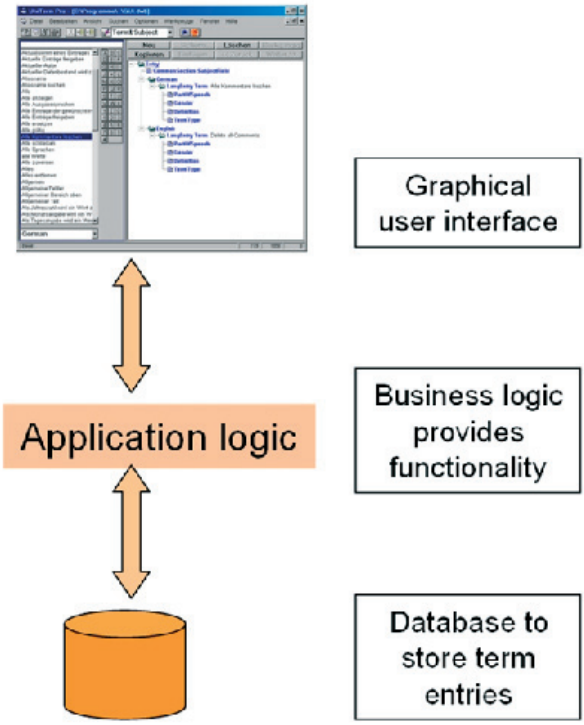

Fig. 1: Typical software architecture for terminology management systems as database applications

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.

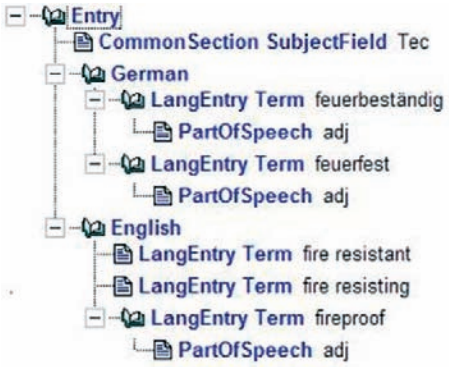

Fig. 2: Sample entry coding in UniTerm

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

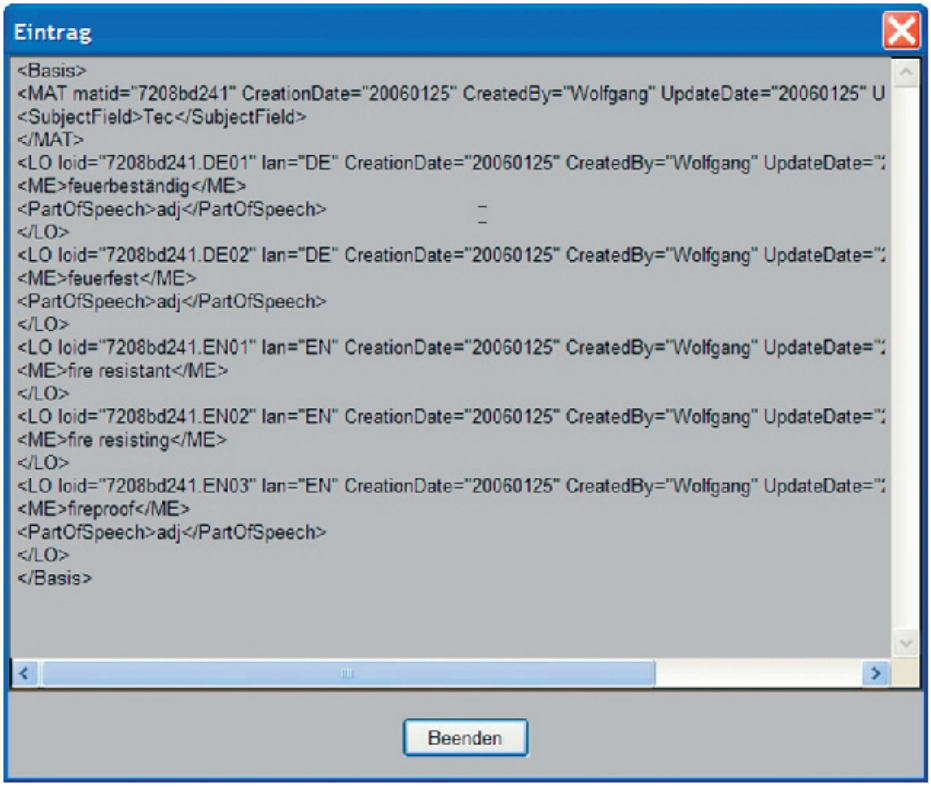

Fig. 3: XML representation of a sample coding in the UniTerm database

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 (= **T**erm**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.

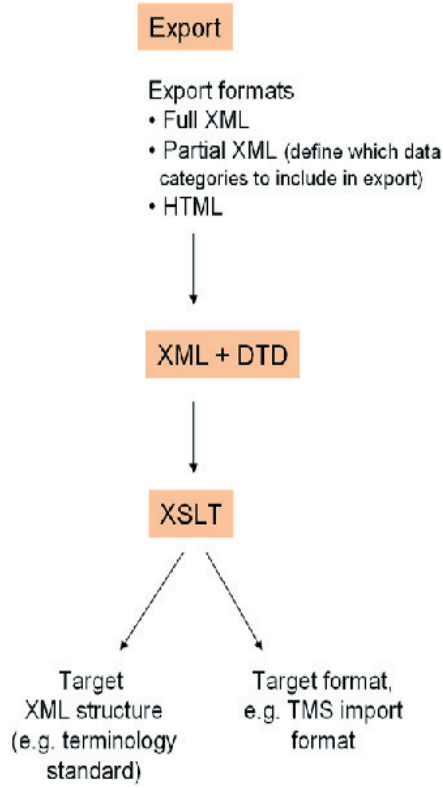

Fig. 4: Model terminology exchange process from XML format

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".