OpenNetTerminologyManager- a Web and Standards based OpenSource Terminology Management Tool

Klemens Waldhör*

* Friedrichstr. 17, 90574 Roßtal, Germany, dr.klemens.waldhoer@waldhor.com

Abstract

OpenNetTerminologyManager is a privately started Open Source project which aims at developing a freely available pure web based concept terminology management system. It runs with any browser supporting JavaScript. The server side requires MySQL, Apache Web Server and Perl. The system is currently available through sourceforge.net at http://openwebterm.sourceforge.net. OpenNetTerminologyManager supports different terminological models. A version which is based on MARTIF has been implemented.

1. Introduction

Through the years the world has seen the attempt to establish several different terminology standards starting from MARTIF, Geneter to TBX, XLT (SALT), TMF (ISO 16642) and so on. The author himself was part of one of the older efforts which started 1990 where within the MULTILEX project a first try was made to create a standard exchange description for areas like mono- and multilingual dictionaries, machine translation etc. The basic idea there was to use SGML as the description language. This was followed up in projects like EAGLES, Otelo (OLIF) etc. In parallel other attempts have been made like Geneter. Sometimes one is really puzzled how creative the terminology community is in inventing new ideas and standards. Often it is really hard to follow what is going on. This is the one side of the coin. On the other side the industry uses "**quasi standards**" like the export format used in MultiTerm™ from Trados™. Several products of competitors like TermStar™, UniTerm™ and others provide import and export features from and into the MultiTerm™ format, simply because MultiTerm™ is the market leader in this area. Otherwise getting into this application field for new systems is nearly impossible as most customers either use MultiTerm™ or at least provide their data in this format.

Interestingly enough Open Source terminology software was never really part of the terminology game, in contrast to other areas like web servers where open source software like Apache is the dominating software (60 % of the world web server market). If one searches for "open source terminology management" in Yahoo and inspects the returned results in detail there are only two other relevant matches, the ForeignDesk and OpenGALEN match. In the last half year Lionbridge has made its software **ForeignDesk** available through open source. Another notable effort is **RosettaWerks** which deals implementing a set of tools for the localisation process.

But what is really missing is a terminology tool which is available on several operating systems (not just Windows™) and can be used through the web and itself is built on free available software. This is not the place to discuss the advantages of the open source model. A lot of discussion is going on this area, but I just want to add that one clearly has to distinguish the open source model from models which are offered by software suppliers where one can get the executables for free, but has no access to the source code. Several providers of terminology software supply down-graded or full versions of their tools – mainly viewers - e.g. UniLex™ from Acolada GmbH, but this does not bring any advantage to the user as he still relies on the provider to fix bugs etc. In addition it is hard to check if there are any hidden traps in the software. As professional terminology management contains company or customer information security aspects and the ability to check this will be an important aspect of choosing a system in the future. Based on this observations – and being also a fan of the open source community - I started developing a terminology management software which should fill this gap.

2. OpenNetTerminologyManager Terminology Model

The basic idea of the system architecture is the capability to support **different terminology models**. The user should have the option either to create his own model or to adapt an existing model by sub-classing it or adding his own fields. It should also be possible to keep track with on-going changes in the standardisation community. This has been realised in the system in the following way: attributes (elements) of the terminology model are not directly mapped to database tables, but this information is kept in a specific column where the structure can be freely defined. The actual mapping of these content of this column to attributes is defined in **model files**. Each database represents one model. The advantage of this approach is a) that it keeps the number of databases tables to a minimum, b) as a result the system is quite fast in searching and reading entries and c) adaptations of attributes can be made easily.

The basic OpenNetTerminologyManager approach is **concept oriented** as it used in most modern terminology systems. In this approach a concept corresponds to one meaning of a word. The language specific parts of a concept are called "**language terms**" or simply "**terms**". Each concept is tagged with an unique identifier, while each term related to the concept uses the concept identifier plus a language identifier and an internal term counter as identifier.

Example: The German term "**Birne**" (three meanings: Glühbirne, Frucht, Kopf = bulb, pear, nut) will be represented by creating three concepts (Figure 2):

a) one with the meaning of "Frucht = fruit" and

b) one with the meaning of "Glühbirne = bulb" and

c) one with the meaning of "Kopf = head".

The **kernel** of OpenNetTerminologyManager consists of several tables:

a) A **MONOTERM** table which holds all relevant information for a term including term attributes

b) A **MULTITERM** table which links entries in the MONOTERM table to a concept and also also stores concept related attributes.

c) A **DETAILS** table which contains links from attributes to terms and concepts. This table is only used to optimise the speed when searching with attributes.

d) A **LINK** table which establishes links between either concepts or term (e.g. in order to express a relations like "synonym").

Different terminology models are now mapped to the kernel model in a **model file**. This model file defines:

The **names** (e.g. "Gender") to be used for the **attributes** of the terminology model into an internal name. This association differentiates between concept related and term related attributes.

The **values** and **forms** to be associated with a such **names**. As an example associate the attribute "Gender" with three possible values ("male", "female", "neuter") and display them in the browser as a select box.

Table 1 shows a simple section for the MARTIF model. Models can further be differentiated into two classes: "**full models**" and "**sub-models**". A sub-model is defined as a subset of attributes of a full model. This is mainly necessary if for a given model (e.g. MARTIF) only specific attributes should be shown or if specific restrictions may apply for attribute values. The system contains some additional fixed attributes like the owner of the concept, read and write accesses etc.

3. OpenNetTerminologyManager Features

The following functions are currently supported:

- Constraints between attributes can be realised with JavaScript
- New models and sub-models can be created by the user (see Figure 1).
- Attributes can be defined by the user.
- Different types for attributes like option fields, text fields, select etc. are supported.
- Multiple databases; multi-user read/write support (locking at concept level). Different right combinations can be used. Databases are either private (with user and password protection) or public.
- Partial Unicode support. Unicode characters above Ascii 255 are stored as SGML entities in the database. This will be removed once MySQL supports directly UTF8 or a similar Unicode encoding scheme. Languages like Arabic, Chinese, Japanese etc. can be used through this approach. Once a Unicode implementation of MySQL is available this representation will be changed to an internal Unicode character set.

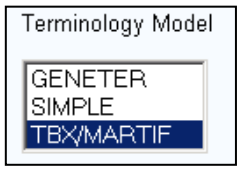


Figure 1: Models

Currently one terminology model based on MARTIF has been (partially) implemented. It normalises the XML definitions into the relational (table based) approach defined above. Others like Geneter are under way.

```
opwdetail40=Grammatical Gender
opwdetail41=Term Type
opwdetail44=Grammatical Number
…
 <tr>
   <td>
     <fieldset>
       <legend>Grammar</legend>
       <table>
         <tr>
           @tdopwdetail43=Part Of Speech?10<select!noun|verb|adjective|other
           @tdopwdetail40=Grammatical Gender?10<select!na|feminine|masculine|neuter|other
         </tr>
         <tr>
           @tdopwdetail41=Term Type?10<select!...|variant
           @tdopwdetail46=Valency?10<input
         </tr>
         <tr>
           @tdopwdetail44=Grammatical Number?10<select!na|dual|mass|other|plural|singular
           @tdopwdetail45=Animacy?10<select!animate|inanimate|other
         </tr>
       </table>
     </fieldset>
   </td>
 </tr>
```

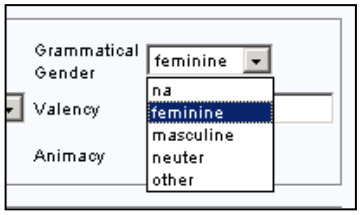


Table 1: OpenNetTerminology Manager GUI description

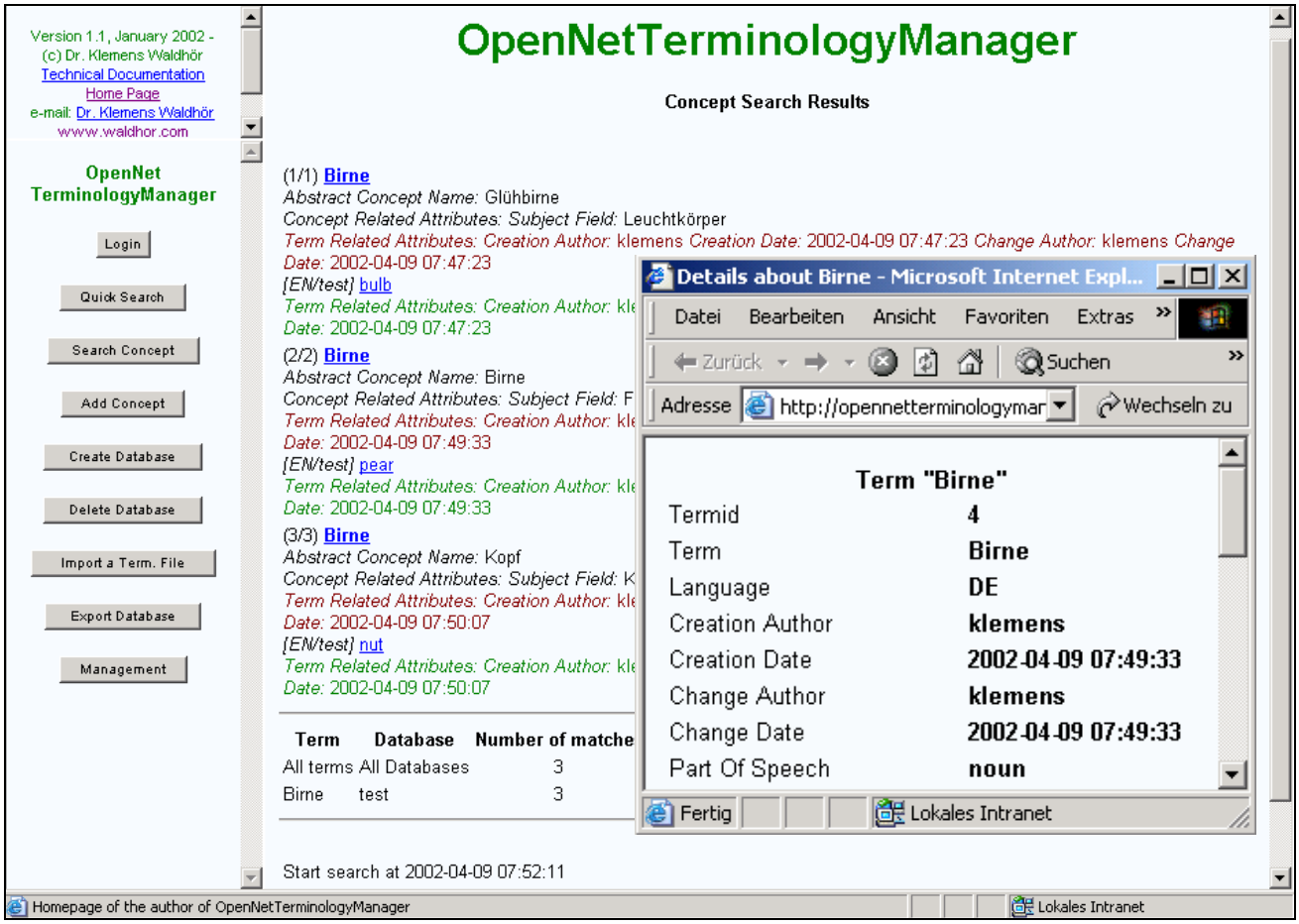
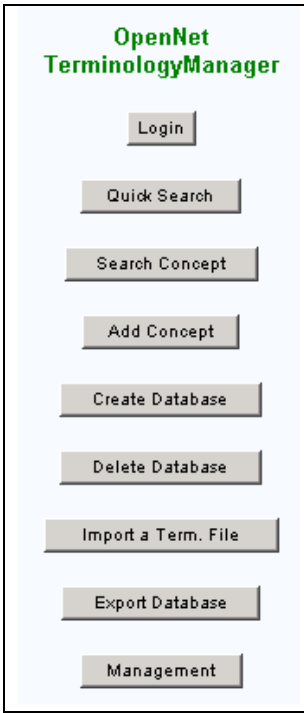


Figure 2: OpenNetTerminology Manager User Interface



Login allows the user to define default values like his preferred databases, languages, attributes to be displayed for a model, how search results should be displayed etc.

Quick Search offers a simplified search mode which simply looks up the database for a specified term independently of the language.

Search Concept supports more detailed search options like languages to be searched and displayed etc. Through this item the user also can edit concepts in the database.

Add Concept adds new concepts to the database.

The user can create new databases using **Create Database** where he also specifies the model and access rights for the database.

Databases can be deleted using **Delete Database**.

Import Term File allows existing terminology files to be imported in various formats supporting double detection during import,

while **Export Database** exports databases into various formats.

Management is mainly intended to give an overview of current system settings and databases. It also supports recreating the database structure.

Figure 3: OpenNetTerminologyManager Commands

4. The Basic User Interface of OpenNetTerminologyManager

Figure 2 shows the basic web based user interface. It consists of a main window where the results of queries etc. are shown and a navigation window (left). Optionally additional concept or term related information can be displayed in a separate browser window. Figure 3 describes the basic functions of the navigation window.

Concepts can be edited by first searching them with the **Search Concept** function and using the "Edit Mode" (not choosing "Dictionary View" option). See figure 4. Results are then displayed in a tabular like format (figure 5). Clicking on "Edit" will then display the full entry (figure 6) in an editable format. Results can also be displayed in a "Dictionary View" mode (figure 7). In this mode concepts found with the same name for a given language may optionally be collapsed into one output entry. This displays the entry in a similar way as they are show in printed dictionaries. Depending on the user search result display settings attributes will be displayed either directly in the main window as part of the entry or the term name is realized as a hyperlink and when clicking on it is displayed later in a separate browser window (figure 2). In addition the user can configure for each database which attributes should be shown. The query itself supports various search options like full text search, regular expressions, the LIKE operator etc.

5. Software requirements

OpenNetTerminologyManager requires the following software components: Perl > 5.0 (with some additional modules installed), Apache Web server or a compatible server, MySQL and a JavaScript enabled Web Browser. Tests have been done with Internet Explorer 5.0, 6.0®, Netscape® and Opera® . The system has been tested both on Windows (NT® and 2000®) and LINUX.

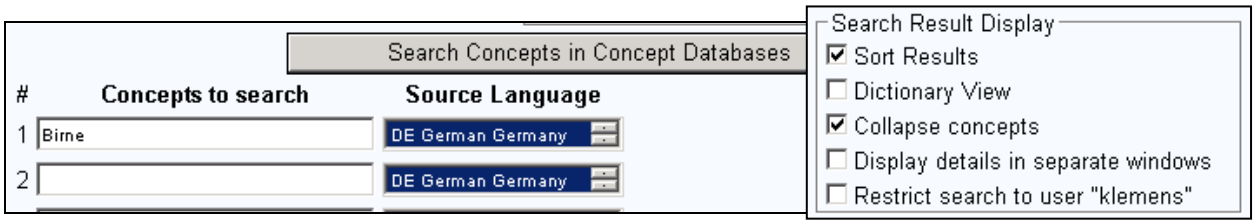


Figure 4: Searching concepts

No	ID / Concept	Source Language DE	Translation Term	Language	Database	Operation
1/1	316071404	Birne	nut	EN	meine	Edit Delete Copy Concept
2/2	284021368 Glühbirne	Birne	bulb	EN	meine	Edit Delete Copy Concept
3/3	312237593	Birne	pear	EN	meine	Edit Delete Copy Concept

Figure 5: Searching result display

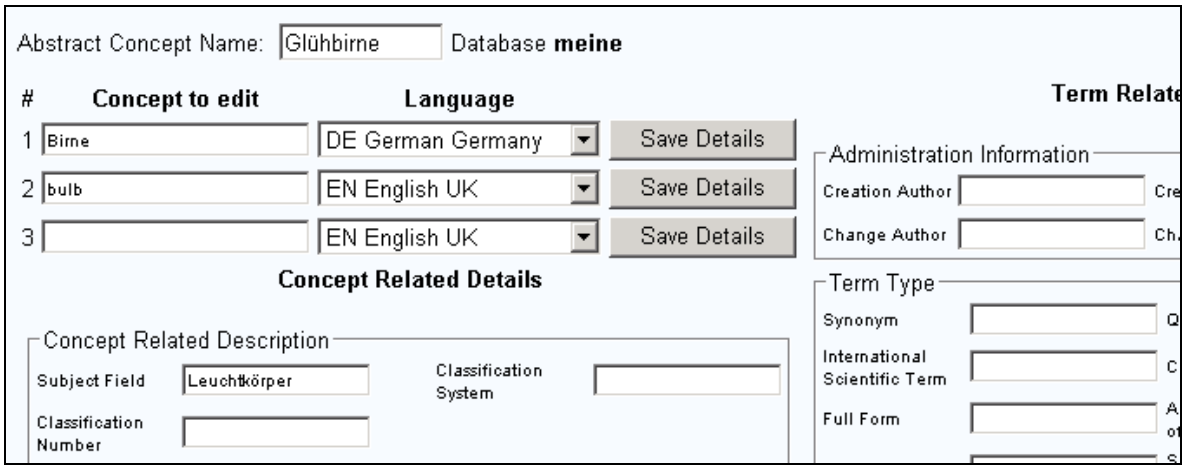


Figure 6: Editing concepts

Figure 7: Dictionary View display with no attributes displayed searching for "Planet%"

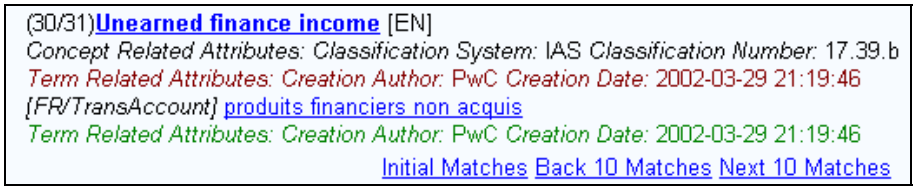


Figure 8: Result of a TransAccount terminology database full text query (searching for the term "finance") with attributes displayed.

6. Application Scenario

The TransAccount project (MLIS 5016) deals with the need for a multilingual translation system allowing the translation and interpretation between the annual accounts of a member state of the European Union (France) and IAS (International Accounting Standards) statements. Within this project the XBRL (eXtensible Business Reporting Language) IASCF taxonomy has been translated from English to French by one of the partners. The resulting 2000 concepts have been imported into a TransAccount terminology database. In addition about 2000 other general financial terms have been converted from a Geneter based format which have been produced by another partner at the start of the project. An example of the results of a query is shown in figure 7.

7. Next Steps

An important feature which is currently under development is an advanced link concept. This link concept will not only support links in the way as TBX defines them but will allow to create complex typed links between concepts and terms and databases. This will allow the user to search the databases not only as a simple term-lookup tool but to browse through it in a kind of semantic net and to find related concepts.

A concept is also developed which supports "similarity queries". It is intended to introduce a "stemming based index" by applying the Porter stemming algorithm to terms for some languages automatically (Porter, 1980). Other developments concern additional import / export formats and simplified form handling for attributes.

As there are several opens source project on mapping xml to relation databases on the way (e.g. XML-DBMS) I am currently also looking into replacing the internal structure of the database by a full xml database approach. This will heavily depend on the access speed compared to the current implementation.

8. References

Acolada. http://www.acolada.de

ForeignDesk. http://sourceforge.net/projects/foreigndesk/

OLIF. http://www.olig.net

OpenGALEN. http://www.opengalen.org/

OpenNetTerminologyManager. http://openwebterm.sourceforge.net

Porter, M.F., 1980. An algorithm for suffix stripping, *Program, 14 no. 3, pp 130-137, July 1980*

Rosettawerks. http://rosettawerks.sourceforge.net/Default.php

Sourceforge. http://sourceforge.net/

Star AG. http://www.star-ag.ch/eng/software.html

Trados. http://www.trados.com

TransAccount: http://www.transaccount.org

XML-DBMS. http://www.rpbourret.com/xmldbms/index.htm

Waldhör, K., Tesniere, B., 2002. Multilingual Terminology Database, *MLIS 5016 TransAccount Report*.

XBRL. http://www.xbrl.org

9. Acknowledgements

Thanks has to be given at this place to SourceForge which provides an excellent – and free – way to make open source projects available through the web.