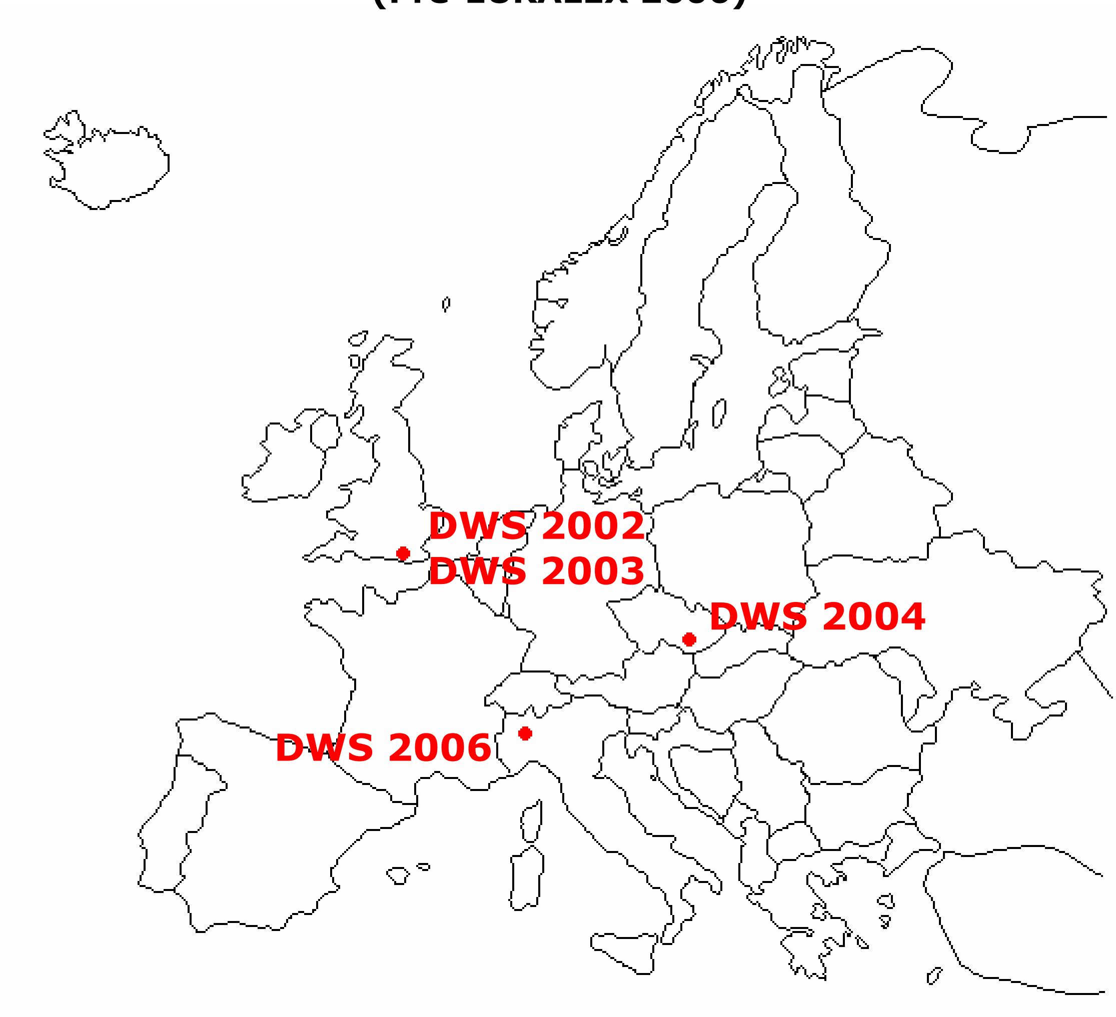
DWS 2006: Proceedings of the Fourth International Workshop on Dictionary Writing Systems

Tuesday 5th September 2006

Turin, Italy

(Pre-EURALEX 2006)



Workshop organisation:
Lexical Computing Ltd., U.K.
Faculty of Informatics, Masaryk U, Brno, Czech Republic

Proceedings editor:
Gilles-Maurice de Schryver

DWS 2006: Proceedings of the Fourth International Workshop on Dictionary Writing Systems

Tuesday 5th September 2006

Turin, Italy
(Pre-EURALEX 2006)

Organised by:

- Lexical Computing Ltd., U.K.
- Faculty of Informatics, Masaryk U, Brno, Czech Republic

Edited by:
Gilles-Maurice de Schryver

ISBN 0-620-36890-X
Pretoria: $(SF)^2$ Press

Table of Contents

Word from the Chair **7**

Programme **9**

Papers **11**

The IEL dictionary management system of Estonian
— Margit Langemets, Andres Loopmann and Ülle Viks 11

New clients for dictionary writing on the DEB platform
— Aleš Horák, Karel Pala, Adam Rambousek and Pavel Rychlý 17

From an online dictionary to an online dictionary writing system
— Andrea Abel and Stefano Bracco 25

Software demo: The terminographic processor MyTerMS
— Igor Kudashev and Irina Kudasheva 35

The users and uses of TshwaneLex One
— Gilles-Maurice de Schryver and David Joffe 41

CULLER – A user-friendly corpus query system
— Elzbieta Dura 47

Particulars of Presenters and Participants **53**

Word from the Chair

A dictionary writing system (DWS) is a piece of software for writing and producing a dictionary. It might include an editor, a database, a Web interface and various management tools (for allocating work etc.). It operates with a dictionary grammar, which specifies the structure of the dictionary.

The workshop is relevant for:

- dictionary project managers;
- lexical database users and developers;
- lexicographers;
- students of lexicography, lexicology, computational linguistics.

The workshop follows similar successful events in Brighton, U.K. in 2002 and 2003, and Brno, the Czech Republic in 2004.

— Adam Kilgarriff

Programme

Venue: *Room 4, Palazzina Einaudi, Università degli Studi di Torino, Lungo Dora Siena 68, Turin, Italy*

Date: *Tuesday 5th September 2006*

13:00–13:45	Registration
13:45–14:00	Welcome, opening comments
14:00–14:30	Margit Langemets, Andres Loopmann and Ülle Viks **The IEL dictionary management system of Estonian**
14:30–15:00	Aleš Horák, Karel Pala, Adam Rambousek and Pavel Rychlý **New clients for dictionary writing on the DEB platform**
15:00–15:30	Andrea Abel and Stefano Bracco **From an online dictionary to an online dictionary writing system**
15:30–16:00	Coffee break
16:00–16:30	Igor Kudashev and Irina Kudasheva **Software demo: The terminographic processor MyTerMS**
16:30–17:00	Gilles-Maurice de Schryver and David Joffe **The users and uses of TshwaneLex One**
17:00–17:10	Closing

Reserve paper: *Elzbieta Dura*
CULLER – A user-friendly corpus query system

The IEL dictionary management system of Estonian

Margit Langemets, Andres Loopmann & Ülle Viks

Institute of the Estonian Language, Tallinn, Estonia
E-mail: {margit.langemets,andres.loopmann,ylle.viks}@eki.ee
Web: http://www.eki.ee/index.html.en

Abstract The demo presents a Dictionary Management System (created at the Institute of the Estonian Language) containing tools for lexicographers to compile, edit and layout dictionaries. The independent components of the System are: (a) dictionary databases in XML format, and (b) software for dictionary management. The System has been successfully implemented by compiling and editing Volume 4 of the five-volume Estonian–Russian Dictionary (EST-RUSS 1997–) and by editing of the Orthological Dictionary (OD 2006). The System is meant to be developed into an interactive lexicographer's working environment, incorporating other language resources as well as language software. The System exists in a local as well as a Web version.

1. Background

Around twenty electronic versions of dictionaries have been compiled by different teams of lexicographers and language technologists at the *Institute of the Estonian Language* (henceforth IEL) since 1978. These lexical resources differ substantially in their realisation, as technical equipment as well as overall knowledge have changed enormously over the past quarter of a century.

Like many other countries we have followed the scheme of starting from typographical or presentational markup – implicit from the point of view of the inner structure and function of the text – then moving on towards an explicit structural view, firstly, using linear descriptional markup, and later on, generic markup, which enables one to describe the structure of the document in greater or less detail and any features one desires to encode, and, of course, to process the document algorithmically, as well as its final publication. The methods of lexical encoding are described in Langemets (2000; 2002), an overview of earlier electronic dictionaries of Estonian is given in Viks (1990).

The first attempt to associate structural control with direct compiling and input of a dictionary was made in the early 1990s, when the manuscript of Volume 1 of the five-volume Estonian–Russian Dictionary (EST-RUSS 1997–) had been completed. The lexicographer was guided by a special computer program to fill in the content of different structural elements of the entry in strict order, following the rules for the specific context. For example, after entering the headword, one could choose either sense indication, definition or translation, after the translation equivalent one had to present Russian grammatical information, etc. So, the structure of the document was checked against the rules, but the structural description was still linear, not hierarchical, thus ignoring the real picture of the entry. In addition, during the many years of compiling the manuscript the dictionary program, which was not user-friendly at all, had gone extremely out of date.

In 2005 we started implementing new in-house software – the IEL Dictionary Management System –, aiming at becoming more universal, Web-based, and user-friendly (Loopmann *et al*. 2006). The two independent components of the System are: (a) dictionary databases, and (b) software for dictionary management.

2. Dictionary databases

The format for presenting dictionary data is XML, accepted widely as a standard for describing dictionaries and other language resources (XML 1.0; Calzolari *et al*. 2001). By means of XML the structural elements (headword, senses, examples, etc.), covering the whole content of the dictionary, are encoded. The physical and logical structure of each dictionary is defined by the schema. The result (i.e. the compiled entries) is validated against the schema, both at user option and generally, to guarantee the well-formedness of the entire document.

The XML markup may be generated in different ways:

- manually, using an ordinary text editor;
- semi-automatically, transforming and deriving data from other markup styles;
- automatically, using the Dictionary Management System.

Only few databases have been marked manually. These have been small, temporary interim forms for presenting some kind of original material (usually created with the help of *Microsoft* Word macros), incorporated into other lexicons as soon as possible.

Semi-automatic transformation of dictionary data into XML format is the most used method (both now and in the nearest future), as all the major dictionaries started in the 1980s or earlier have been fed into the computer using linear descriptional markup. By now the procedure of transformation has been completed for two voluminous traditional dictionaries. In 2005 the editorial team of the Estonian–Russian Dictionary tested the new dictionary system, then being the pioneers of using it in their everyday work for compiling and editing the last two volumes of the dictionary (out of five, with a total of 80,000 headwords). The second ‘reformatted’ dictionary is the monolingual Orthological Dictionary (OD 1999), the supplemented edition of which is to be published in 2006 (OD 2006).

The earlier electronic versions of dictionary texts may be quite specific, each demanding a lot of work. In some cases the entries have to be restructured to fit into the XML schema as the simplest possible (the more complex the structure, the more problems with using the system). The effort of standardising, however, is worthwhile when a dictionary text is to be reused, for example as a basis for new dictionaries, or for supplementing the same dictionary, or as a possible component of different language technology applications.

Fully automatically, using the Dictionary Management System, are and will be compiled all new original dictionaries initiated at the Institute, the first of which is the ongoing national project Estonian-X Dictionary (40,000 headwords) aiming to provide the source language (L1) for different medium-sized bilingual dictionaries.

3. The management system

The Dictionary Management System (DMS) enables dictionaries to be stored in XML format. Alongside lexicographical functions (compiling and editing dictionary entries) it includes some general functions (Web interface, dictionary layout, etc.). The basic functions and requirements of DMS follow the needs of the lexicographer’s working process and are oriented to making the DMS user-friendly.

3.1. DMS basic functions

1) adding (compiling) a new entry;
2) modifying (editing) an existing entry: adding/changing/deleting elements and attributes;
3) deleting an entry;

4) entry search, based on various features: element/attribute existence, their value; additional options are considering or ignoring non-letter symbols and case sensitivity, use of meta-characters;
5) dictionary layout: viewing entry, displaying search results;
6) alphabetical sorting of entries;
7) validation of an entry against the XML schema;
8) log of the working process: saving the date, time and username for every entry.

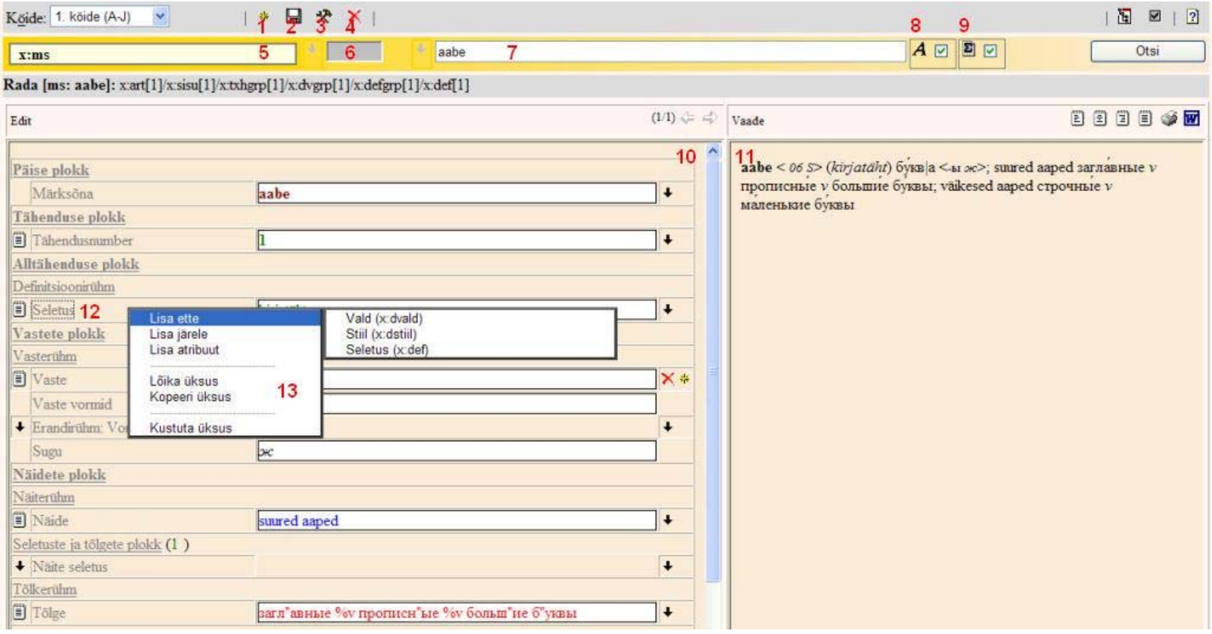
3.2. DMS basic requirements

1) Web-based organisation of the DMS: a dictionary should be edited collectively;
2) workstation software should be based on standards and the components should be easily installable;
3) search of entries should be simple, using a number of different characteristics;
4) allowed operations should be context sensitive: the user need not be aware of the technical characteristics of the XML schemas;
5) an entry should meet the XML schema requirements; if not, the entry cannot be saved;
6) the user changes only text, while the entry view and layout are generated automatically;
7) an XML element and its view in different areas are connected: clicking on an XML element shows it in preview with a different background colour and vice versa;
8) the user's username and date/time are saved automatically.

3.3. DMS technical realisation

On the server side, an Apache Web server is used; on the workstation side, *Microsoft* Internet Explorer 6 or higher is required. The DMS window has three areas: (a) the editing area visualises the structure of the entry and presents every XML element in its own editing box; (b) the preview area visualizes the format of the entry similar to the final look of the printed document; (c) the functional area is for menus, buttons and info.

The following example shows the entry *aabe* 'letter' in the Estonian-Russian dictionary:



Kõide: 1. köide (A-J)

x:ms

aabe

Rada [ms: aabe]: x:art[1]/x:sisu[1]/x:txhgrp[1]/x:dvgrp[1]/x:defgrp[1]/x:def[1]

Päise plokk

Märksõna: aabe

Tähenduse plokk

Tähendusnumber: 1

Alltähenduse plokk

Definitsioonirühm

Seletus

Vastete plokk

Vasterühm

Vaste

Vaste vormid

Erandirühm: Vo

Sugu: ж

Näidete plokk

Näiterühm

Näide: suured aaped

Seletuste ja tõlgete plokk (1)

Näite seletus

Tõlkerühm

Tõlge: загл"авные %v прописн"ые %v больш"ие б"уквы

aabe < *06 S*> (*kirjatäht*) букв|а <-ы *ж*>; suured aaped заглавные *v* прописные *v* большие буквы; väikesed aaped строчные *v* маленькие буквы

On the left side, the editing area (10) is shown, on the right, the preview area (11) is shown. XSLTs (Extensible Stylesheet Language Transformations) are used for generating the editing and preview areas. At the top of the screenshot, the functional area is shown. On the topmost row, buttons for adding a new entry (1), saving an entry (2), renaming an entry (3), and deleting an entry (4) are shown. On the second row, the menu of element names (5) is shown. Input box (6) is meant for attributes text, box (7) for elements text. Checkbox (8) determines the case sensitivity of the search, checkbox (9) determines if non-letter symbols are included in the search. A context menu (13) appears when the element name (12) is right-clicked. By means of context menu, one can add/delete elements/attributes, copy/paste elements, etc. According to the schema context one can add a new element before or after a clicked element. On top of the preview are buttons for entry layout and printing. Entries chosen for layout are displayed in Word.

3.4. Pre-processing of dictionaries

The loading of an existing dictionary into the DMS involves several procedures (XML format is a condition sine qua non):

1) creating the schema of the dictionary;
2) creating two views: one for the editing area (structured view) and another for the preview area (layout view);
3) working out the templates for adding new dictionary entries, new hierarchy groups, etc.

Procedures of creating the schema and the views are automated, and after loading an existing XML-dictionary only tuning is needed. In case a dictionary is to be compiled from scratch the schema is created in tight cooperation with lexicographers: all elements, attributes, groups and their properties are predefined manually.

4. Current developments, problems and perspectives

The IEL Dictionary Management System has been successfully implemented. Volume 4 of the five-volume Estonian–Russian Dictionary (EST-RUSS 1997–) has been completely compiled and edited using the DMS. The database of the Orthological Dictionary (OD 1999) has been ‘reformatted’ for the system, the ongoing work is concerned with the final supplements of the new edition of the dictionary (OD 2006). The work with the biggest dictionary of Estonian – the Defining Dictionary of Estonian (DD 1988–, the manuscript to be finished in 2007, covering 150,000 headwords) – is in the preliminary phase; the lexicographic work with new words (forming a supplement of the DD) has been started using the DMS.

Lexicographers (editorial teams) differ a lot in their attitudes towards changing working habits, or towards machines in general, so it demands a lot of patience and time to teach lexicographers to start accustoming themselves to system-based thinking.

The XML database of the Estonian–Russian Dictionary will be one of the most important lexical bases for creating the Estonian–X Dictionary database (40,000 headwords). So far it has been used as a direct source material for two dictionary projects (Estonian–Latvian and Estonian–French).

The IEL Dictionary Management System is meant to be developed into an interactive lexicographer’s working environment which, being as user-friendly as possible, should enable the editors to focus only on the content of the dictionary. This purpose will be served by incorporating other language resources (text corpora, other dictionaries and databases, etc.) as well as language software (automatic morphology, derivation and compounding, statistics, etc.). This will enormously facilitate automatic generation of morphological data (part of speech, inflectional type, etc., cf. Viks 2000), importing appropriate linguistic data (dependency relations, definitions, style labels, etc.) from other dictionaries, importing appropriate text examples from corpora and/or the Web, etc.

The IEL Dictionary Management System exists in a local as well as a Web version, and besides managing lexicographic data, one may well imagine it adapted for managing all kinds of structural data.

Acknowledgements

The work has been supported by EKRK47, ETF5969 and SF0052488s03.

References

A. Dictionaries

DD 1988– = *Eesti kirjakeele seletussõnaraamat 1–25* (26) [The Defining Dictionary of Estonian]. Tallinn: Eesti Keele Sihtasutus.

EST-RUSS 1997– = *Eesti-vene sõnaraamat 1–3* (5) [Estonian–Russian Dictionary]. Tallinn: Eesti Keele Sihtasutus.

OD 1999 = Erelt, T. (ed.). 1999. *Eesti keele sõnaraamat ÕS 1999* [Orthological Dictionary]. Tallinn: Eesti Keele Sihtasutus.

OD 2006 = Erelt, T. (ed.). *forthcoming* in 2006. *Eesti keele sõnaraamat ÕS 2006* [Orthological Dictionary]. Tallinn: Eesti Keele Sihtasutus.

B. Other literature

Calzolari, N., R. Grishman & M. Palmer (responsible authors). 2001. *Survey of Major Approaches Towards Bilingual/Multilingual Lexicons*. ISLE Computational Lexicons Working Group, Deliverable D2.1-D3.1, February 2001.

Langemets, M. 2000. Leksikaalse info kodeerimine. In Hennoste, T. (ed.). 2000. *Arvutuslingvistikalt inimesele* (Tartu Ülikooli üldkeeleteaduse õppetooli toimetised 1): 101–126. Tartu: Tartu Ülikooli Kirjastus.

Langemets, M. 2002. Eesti Keele Instituudi elektrooniline keelevara. *A&A* 5: 39–46.

Loopmann, A., K. Sein & Ü. Viks. 2006. Sõnastike haldussüsteem Eesti Keele Instituudis. In Koit, M., R. Pajusalu & H. Õim. (eds.). 2006. *Keel ja arvuti* (Tartu Ülikooli üldkeeleteaduse õppetooli toimetised 6): 246–258. Tartu: Tartu Ülikooli Kirjastus.

Viks, Ü. 1990. Sõnastike andmebaas: milleks, mis ja kuidas. In Ross, J. (ed.). 1990. *Arvutuslingvistika sektori aastaraamat 1988*: 167–175. Tallinn: Keele ja Kirjanduse Instituut.

Viks, Ü. 2000. Tools for the Generation of Morphological Entries in Dictionaries. In *Proceedings of the Second International Conference on Language Resources and Evaluation, Athens, Greece, 31 May – 2 June 2000*: 383–388.

XML 1.0 = *Extensible Markup Language (XML) 1.0 (Third Edition)*. W3C Recommendation 04 February 2004. Available from `http://www.w3.org/TR/2004/REC-xml-20040204/`

New clients for dictionary writing on the DEB platform

Aleš Horák, Karel Pala, Adam Rambousek & Pavel Rychlý

Faculty of Informatics, Masaryk University, Brno, Czech Republic
E-mail: {hales,pala,xrambous,pary}@fi.muni.cz
Web: http://www.fi.muni.cz/

Abstract In this presentation, we offer an overview of the new clients based on the XML database system called DEBII. Thanks to the versatile nature of the XML format this platform enables us to develop various applications, namely the management (editing, browsing and other functions) of the electronic readable dictionaries, WordNet-like lexical databases as well as ontologies for Semantic Web applications. First, we characterize the main features of the whole DEBII dictionary writing platform, and then the implementation strategies of both server and client part of the DEBII platform are briefly described. Second, we present the following tools/clients: 1. DEBVisDic – which allows to handle different lexical resources and can be used as an appropriate tool for future standardisation of WordNet-like databases, 2. PRALED – a client for building the Czech Lexical Database, 3. DEBDict – a browser for parallel viewing of several electronic dictionaries, 4. DEB CPA browser and editor – a client for development of corpus patterns of verbs, and 5. DEB TEDI tool – a client for building the Czech Terminological Dictionary. For each of the mentioned DEB clients we give their main features and briefly describe their functionality.

1. Introduction

There is a need to handle various lexical resources that take the form of wordnets, ontologies, valency lexicons, framenets and others. For this purpose researchers seek software systems that are able to store dictionary-like data using XML as the core element. Many dictionary publishing houses operate large systems with the complex functionality of so-called lexicographic stations that manipulate XML (*DPS Longman* (McNamara 2003), *TshwaneLex* (Joffe & De Schryver 2004), *iLEX* (Erlandsen 2004) or *Shoebox*). However, these and similar tools are not always able to efficiently merge and manipulate resources obtained from data-driven NLP applications. Therefore, they cannot provide a universal environment for lexical database management as well as semantic networks and ontologies. They also represent rather large systems that are quite complex which is not always an advantage. Last but not least, some of them are not so cheap.

We decided to build a DEBII platform on which the individual clients can work – in our view this solution is quite modular and flexible since the clients can be adapted for the particular purpose in a short time. One of the reasons for this solution is the fact that some well-known lexical resources in the NLP field take the form of semantic networks – the best examples are the Princeton *WordNet* (Fellbaum 1998), multilingual *EuroWordNet 1* (2 projects, 1998-99), and also the *BalkaNet* project (2001-4) in which the wordnets for 13 languages have been developed (English, Dutch, Italian, Spanish, French, German, Czech, Estonian, Bulgarian, Greek, Romanian, Serbian and Turkish).

In the course of the BalkaNet project's work, the specialised software tools for browsing and editing wordnets have been designed and implemented, without which the job could hardly have been performed – the editor and browser VisDic (Horák & Smrž 2004). The tool has its limitations – it was designed as a local tool only and its flexibility is rather limited.

2. The main features of a common XML platform

The DEB platform (DEBII, i.e. its second version) follows a strict client-server architecture. The actual development of applications within the DEB platform can be divided into the server part (the server side functionality) and the client part (graphical interfaces with only simple functionality). The server part is built from small parts, called servlets, which allow a modular composition of all services.

The clients communicate with servlets using HTTP requests in a manner similar to the popular concept in Web development called AJAX (Asynchronous JavaScript and XML, Rosenfeld & Morville 1998). The data are transported (using plain HTTP) in RDF, generic XML or plain text formats, or they are marshalled using JSON (JavaScript Object Notation) data structure encapsulation.

The actual data storage backend on the server side is provided by the Berkeley DB XML, which is a native XML database providing XPath and XQuery access into a set of document containers. The metadata are stored in the widely-used Berkeley DB embedded database which runs on many systems and devices ranging from Linux and Windows operating systems to mobile phones. The Berkeley DB XML comes in the form of a C++ library with interfaces to many scripting languages.

Since the client applications are mostly oriented to the GUIs (graphical user interfaces), we have decided to adopt the concepts of the *Mozilla* Development Platform (Boswell *et al*. 2002). The Firefox Web browser is one of the many applications created using this platform.

The Mozilla Cross Platform Engine provides a clear separation between application logic and definition, presentation and language-specific texts. The application design is simple and allows for the possibility of concurrent work by different team members which leads to significant time savings.

The main ‘programming language’ used for the GUI design of the DEB clients is called XUL (XML User-interface Language, pronounced ‘zool’). XUL is a user interface description language based on XML. It allows for the relatively simple creation of cross-platform applications with the possibility of easy customisation of design, texts and localisation. XUL itself is aimed only at creating a user interface (e.g. windows, buttons or toolbars), but it incorporates a wide range of standard technologies:

- Cascading Style Sheets (CSS) for the visual style of the application;
- JavaScript as a programming language for simple application logic;
- Document Object Model (DOM), XSLT and XPath to work with HTML and XML documents;
- DTD for easy localisation;
- RDF as data source.

2.1. Why client-server architecture?

In the client-server environment, the server provides different interfaces using the same data structure and these interfaces can be reused by many client applications. For example, several client applications are using the same interface to query XML dictionaries (with different underlying structures).

One of the main benefits of developing a new dictionary tool on the DEB platform is the homogeneity of the data structure and presentation. If the tool developer commits a change in the data presentation, this change will automatically appear in each client software. And of course, any data flaws discovered can be instantly corrected: there is no need to change the client software or provide new data files to each client.

The data sources can even be implemented with different structures, which the server transforms seamlessly to a homogeneous form, which is then provided to client applications.

2.2. The clients-users' interfaces

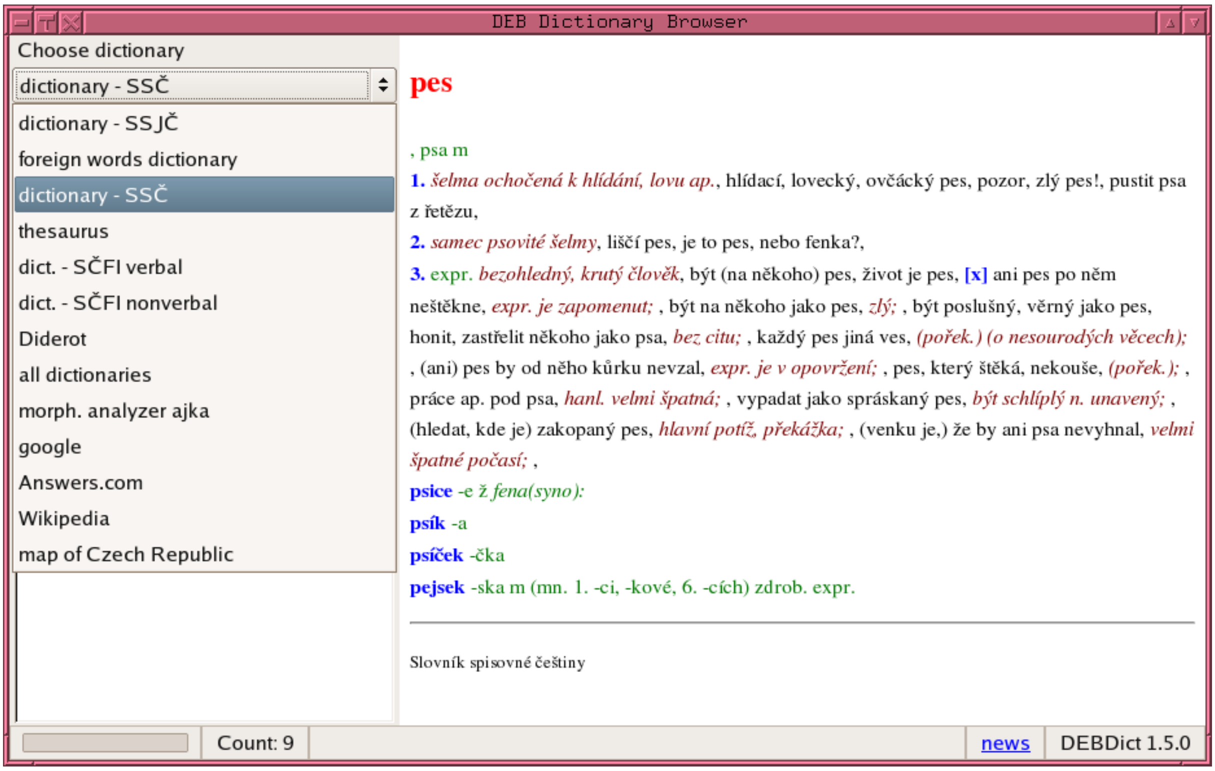
The DEB clients are written in XUL and JavaScript and integrate with the Mozilla Firefox Web browser. This allows us to use both Mozilla's user interface engine and its HTML/XHTML rendering engine as well as built-in components for interaction with the file system on the client computers, XPath interpreter, RDF processor, etc.

Due to the feature-rich client architecture the developers may decide whether certain operations should be done on the server or on client parts – for instance XSLT transformation can be done on both sides.

The particular DEB clients that are currently being implemented within the DEB platform include DEBDict, DEBVisDic, PRALED, DEB CPA and DEB TEDI. We will briefly describe each of them in the next paragraphs.

DEBDict – general dictionary browser. This simple DEB client demonstrates several basic functions of the system:

- multilingual user interface (English, Czech, others can be easily added);
- queries to several XML dictionaries (of different underlying structures) with the result passed through an XSLT transformation;
- connection to Czech morphological analyser;
- connection to an external website (Google, Answers.com);
- connection to a geographical information system (display of geographical links directly on their positions within a cartographic map) or any similar application.



pes

, psa m

1. *šelma ochočená k hlídání, lovu ap.*, hlídací, lovecký, ovčácký pes, pozor, zlý pes!, pustit psa z řetězu,

2. *samec psovité šelmy*, liščí pes, je to pes, nebo fenka?,

3. expr. *bezohledný, krutý člověk*, být (na někoho) pes, život je pes, **[x]** ani pes po něm neštěkne, *expr. je zapomenut;* , být na někoho jako pes, *zlý;* , být poslušný, věrný jako pes, honit, zastřelit někoho jako psa, *bez citu;* , každý pes jiná ves, *(pořek.) (o nesourodých věcech);* , (ani) pes by od něho kůrku nevzal, *expr. je v opovržení;* , pes, který štěká, nekouše, *(pořek.);* , práce ap. pod psa, *hanl. velmi špatná;* , vypadat jako spráskaný pes, *být schlíplý n. unavený;* , (hledat, kde je) zakopaný pes, *hlavní potíž, překážka;* , (venku je,) že by ani psa nevyhnal, *velmi špatné počasí;* ,

psice -e ž *fena(syno):*

psík -a

psíček -čka

pejsek -ska m (mn. 1. -ci, -kové, 6. -cích) zdrob. expr.

Slovník spisovné češtiny

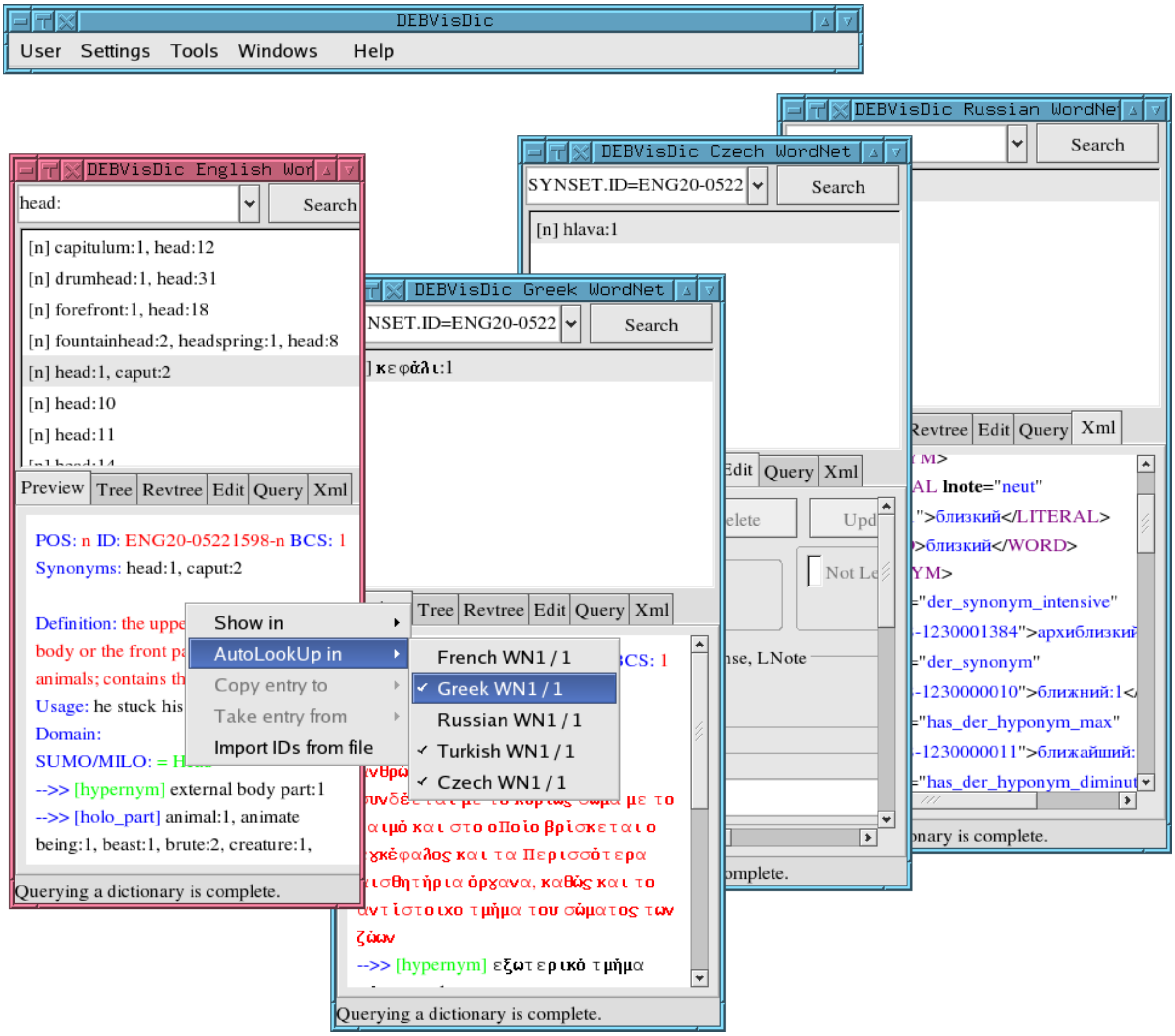
Figure 1: The DEBDict common interface to several dictionaries with different structures

As can be seen in Figure 1, the version of DEBDict that is currently running on our server provides a common interface to seven dictionaries:

- the Dictionary of Literary Czech Language (SSJČ, 180,000 entries);
- the Dictionary of Foreign Words (46,000 entries);
- the Dictionary of Literary Czech (SSČ, 49,000 entries);
- the Dictionary of Czech Synonyms (thesaurus, 23,000 entries);
- two dictionaries of Czech Phrasal Words and Idioms (4,000 and 10,000 entries);
- the Diderot Encyclopaedia (90,000 entries).

As an addition, DEBDict features an interconnection to several Web systems and the geographical system with the list of Czech towns and cities.

DEBVisDic – wordnet editor. DEBVisDic has been conceived as a reimplementation of the previous tool for wordnet semantic networks – VisDic. VisDic already exploits the XML data format thus making the wordnet-like databases more standard and exchangeable. Moreover, thanks to its general configuration, VisDic can serve for developing various types of dictionaries, be these monolingual, translational, thesauri or multilingually linked wordnet-like databases.



head:

[n] capitulum:1, head:12
[n] drumhead:1, head:31
[n] forefront:1, head:18
[n] fountainhead:2, headspring:1, head:8
[n] head:1, caput:2
[n] head:10
[n] head:11

POS: n ID: ENG20-05221598-n BCS: 1
Synonyms: head:1, caput:2

Definition: the uppe
body or the front pa
animals; contains th
Usage: he stuck his
Domain:
SUMO/MILO: = H
-->> [hypernym] external body part:1
-->> [holo_part] animal:1, animate
being:1, beast:1, brute:2, creature:1,

Querying a dictionary is complete.

SYNSET.ID=ENG20-0522

[n] hlava:1

nse, LNote

omplete.

NSET.ID=ENG20-0522

] κεφάλι:1

BCS: 1

νθρώ
συνδέεται με το κυρίως σώμα με το
αιμό και στο οποίο βρίσκεται ο
εγκέφαλος και τα Περισσότερα
αισθητήρια όργανα, καθώς και το
αντίστοιχο τμήμα του σώματος των
ζώων
-->> [hypernym] εξωτερικό τμήμα

Querying a dictionary is complete.

```
M>
AL lnote="neut"
">близкий</LITERAL>
>близкий</WORD>
YM>
="der_synonym_intensive"
-1230001384">архиблизкий
="der_synonym"
-1230000010">ближний:1</
="has_der_hyponym_max"
-1230000011">ближайший:
="has_der_hyponym_diminut
```

onary is complete.

Figure 2: The DEBVisDic interface

The experience with the VisDic tool during the BalkaNet project has been extremely positive (Horák & Smrž 2004) and it was used as the main tool with which all six BalkaNet national wordnets were developed.

Within the development of DEBVisDic we pay attention to the relations between wordnets and the Semantic Web. DEBVisDic uses a new ‘windowed interface’, see Figure 2, that allows a user to arrange the client layout without any limitations. Of course, DEBVisDic contains all the main features that were present in VisDic, like multiple views of multiple wordnets, hypero-hyponymic tree browsing, inter-dictionary linking or synset editing. With the help of the DEB platform reusability, DEBVisDic is supplemented with a number of new features that were so far accessible only as separate tools or resources such as a connection to a morphological analyser (for languages where it is available), language corpora (including Word Sketch statistics), access to any electronic dictionaries stored within the DEB server, or searching within encyclopaedic websites.

The client-server architecture allows for an easy connection of other existing applications to the DEB wordnet server. An example of such an application is a direct interface to the Visual Browser tool (Nevěřilová) that now displays the graphical representation of the semantic network from the same database which is displayed in the DEBVisDic tool.

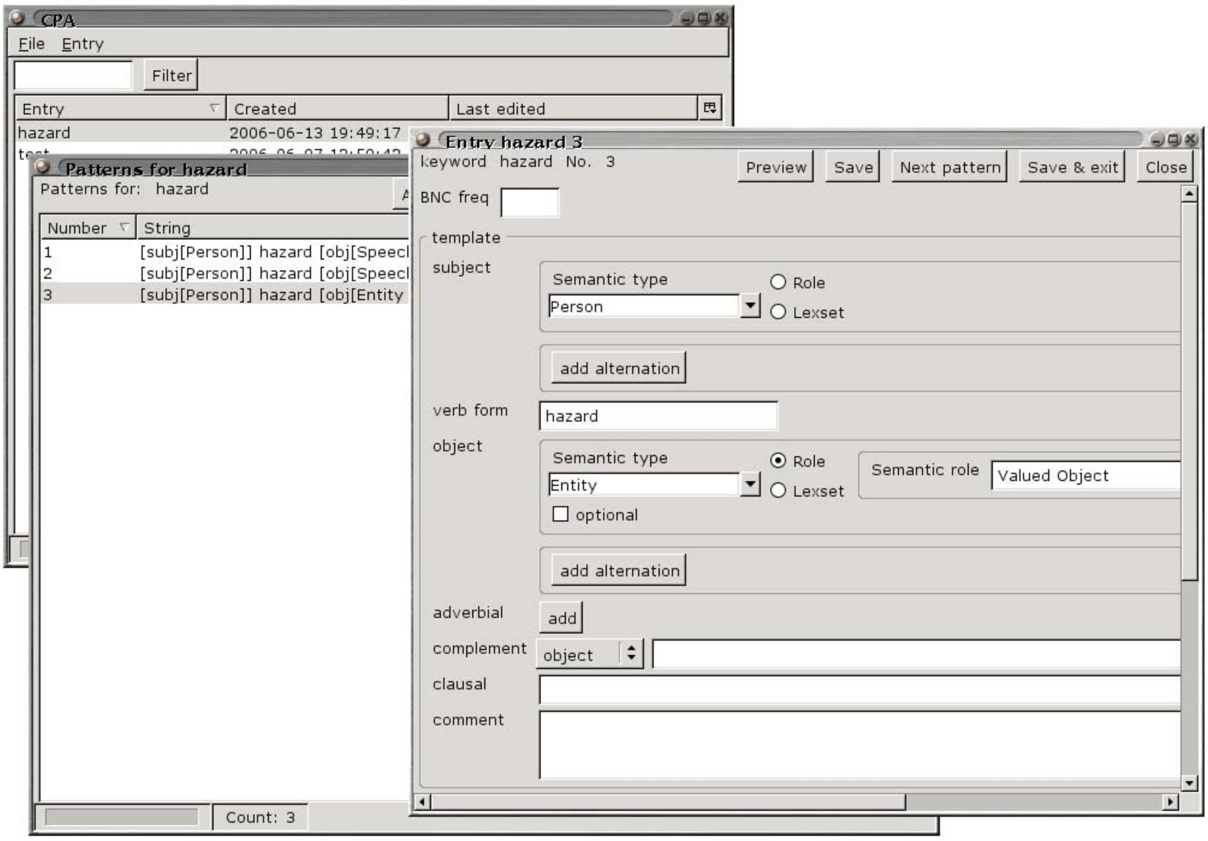
PRALED – Czech Lexical Database tool. PRALED is a browser and editor designed for the development of the Czech Lexical Database, CLD. It serves as the main tool for the preparation of the new comprehensive and exhaustive database of lexicographic information of the Czech language. At present, the user’s part of PRALED is under development at the Institute of the Czech Language, Czech Academy of Sciences, in Prague. Here DEBII is used as a full-blown dictionary writing system platform. Thus the main part of the interface consists of the form for lexicographers who can use it for writing the individual entries. The form contains the following fields:

- variants characterised by the appropriate features;
- morphological information;
- syntactic information in the form of valency frames;
- sense definitions;
- word derivation information;
- semantic relations (hyperonymy/hyponymy, antonymy, cohyponymy, …);
- etymological information (where it is relevant);
- morphological analyser/module, which is not a field but a link.

The forms can be easily linked to the corpus manager Manatee/Bonito and the Word Sketch Engine (Kilgarriff *et al.* 2004). PRALED, in fact, serves as a complete lexicographers’ station.

DEB CPA editor and browser. Corpus Pattern Analysis or CPA (Hanks 2004) is a new technique for mapping meaning to words in text. No attempt is made in CPA to identify the meaning of a verb or noun directly as a word in isolation. Instead, meanings are associated with prototypical sentence contexts. Concordance lines are grouped into semantically motivated syntagmatic patterns. Associating a ‘meaning’ with each pattern is a secondary step, carried out in close coordination with the assignment of concordance lines to patterns.

The CPA editing tool, see Figure 3, displays the list of verb entries, along with the information on who updated which entry when. Each entry consist of several patterns (the number of patterns is not limited) and it is possible to freely modify their order and content. The main part of the tool, the pattern editing window, allows entering and modifying all the information about one pattern.



Entry	Created	Last edited
hazard	2006-06-13 19:49:17	

Patterns for: hazard

Number	String
1	[subj[Person]] hazard [obj[Speecl
2	[subj[Person]] hazard [obj[Speecl
3	[subj[Person]] hazard [obj[Entity

Count: 3

keyword hazard No. 3

BNC freq

subject: Semantic type Person ○ Role ○ Lexset

verb form: hazard

object: Semantic type Entity ● Role ○ Lexset; Semantic role Valued Object; □ optional

adverbial

complement: object

clausal

comment

Figure 3: The DEB CPA tool

The form is very versatile, for instance it allows for the addition of any number of subject/object alternations. The tool is connected to an online resource: it is possible to look up subject and object semantic types in the Brandeis Semantic Ontology (Pustejovsky *et al.* 2006) which is hosted on a Web server at Brandeis University. Examples documenting the pattern are taken from the BNC using a modified version of the Bonito2 corpus manager that is integrated into the DEB CPA tool.

DEB TEDI terminological dictionary tool. The DEB TEDI client is the main tool used for the preparation of a new big terminological dictionary of Czech. This work is a joint project of the Czech publisher *NLN* and Masaryk University. The aim of the project is to build a terminological database consisting of about 250,000 dictionary entries. Several printed dictionaries of different size will be generated from that database.

3. Conclusions and future directions

We have described the current state of development of clients (dictionary tools) based on the DEBII platform. This platform offers a common implementation base for client/server architecture. Thanks to its high modularity, configurability and flexibility it can be easily adapted for various lexicographic tasks. Using this basis, new individual and powerful dictionary writing tools (clients) such as DEBVisDic are implemented.

In our view the DEB platform is being (and will be) thoroughly tested with its clients. For example, the DEBVisDic is currently being prepared for the Dutch Cornetto project and for the Hungarian WordNet project. We are also discussing the possibility of DEBVisDic as the main wordnet tool in the near future, namely for the preparation of the World WordNet Grid (Fellbaum, personal communication, May 2006).

The PRALED client is used at the Institute of the Czech Language, Czech Academy of Sciences, in Prague, as a dictionary writing system for building the Czech Lexical Database which is a large project planned for about six years from now. The goal is to develop a lexical database of contemporary Czech containing approximately 100,000 entries. An important new feature here is that PRALED will be linked to the Manatee/Bonito corpus manager and the Word Sketch Engine.

Acknowledgements

This work has been partly supported by the Academy of Sciences of the Czech Republic under projects T100300414 and T100300419, and by the Ministry of Education of the Czech Republic within the centre of basic research LC05113202.

References

BalkaNet. Available from `http://www.ceid.upatras.gr/Balkanet/`

Boswell, D. *et al.* 2002. *Creating Applications with Mozilla*. Sebastopol, California: O'Reilly and Associates, Inc.

Erlandsen, J. 2004. iLEX – an ergonomic and powerful tool combining, effective and flexible editing with easy and fast search and retrieval (Software demonstration at EURALEX 2004, Lorient, France).

EuroWordNet. Available from `http://www.illc.uva.nl/EuroWordNet/`

Fellbaum, C. (ed.). 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts: The MIT Press.

Hanks, P. 2004. Corpus Pattern Analysis. In Williams, G. & S. Vessier (eds.). 2004. *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004, Lorient, France, July 6-10, 2004*: 87–97. Lorient: Faculté des Lettres et des Sciences Humaines, Université de Bretagne Sud.

Horák, A. & P. Smrž. 2004. New features of WordNet editor VisDic. In *Romanian Journal of Information Science and Technology* 7: 1–13.

Joffe, D. & G-M de Schryver. 2004. TshwaneLex – A State-of-the-Art Dictionary Compilation Program. In Williams, G. & S. Vessier (eds.). 2004. *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004, Lorient, France, July 6-10, 2004*: 99–104. Lorient: Faculté des Lettres et des Sciences Humaines, Université de Bretagne Sud.

Kilgarriff, A., P. Rychlý, P. Smrž & D. Tugwell. 2004. The Sketch Engine. In Williams, G. & S. Vessier (eds.). 2004. *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004, Lorient, France, July 6-10, 2004*: 105–115. Lorient: Faculté des Lettres et des Sciences Humaines, Université de Bretagne Sud.

McNamara, M. 2003. Dictionaries for all: XML to Final Product. In *Proceedings of the XML Conference & Exposition 2003, Philadelphia, Pennsylvania, USA, December 7-12, 2003.*

Nevěřilová, Z. *The Visual Browser*. Available from `http://nlp.fi.muni.cz/projects/visualbrowser`

Pustejovsky, J., C. Havasi, J. Littman, A. Rumshisky & M. Verhagen. 2006. Towards a Generative Lexical Resource: The Brandeis Semantic Ontology (Poster at LREC 2006, Genoa, Italy).

Rosenfeld, L. & P. Morville. 1998. *Information Architecture for the World Wide Web*. Sebastopol, California: O'Reilly and Associates, Inc.

Shoebox. The Linguist's Shoebox. Available from `http://www.sil.org/computing/shoebox/`

From an Online Dictionary to an Online Dictionary Writing System

Andrea Abel & Stefano Bracco

European Academy, Bozen/Bolzano, Italy
E-mail: {andrea.abel,stefano.bracco}@eurac.edu
Web: http://www.eurac.edu/

Abstract Web-based-only dictionaries are rarely so flexible to propose a high granularity level of annotation together with a considerable number of lemmas inserted into them. In many projects we can find a large quantity of lemmas with little information or a small amount of lemmas with a lot of information for each word. ELDIT is a so-called cross-lingual electronic learners' dictionary for German and Italian. In this paper we will propose our approach that will show how to complete and to continually update this web-based dictionary by using an online DWS (dictionary writing system) that is dedicated to people with limited experience in the specific field. The intention of our prototype of a DWS is to help users at different knowledge levels to contribute to the dictionary easily, with an interface accessible from anywhere, using software with an adaptive approach considering the person's skills, experience and data selection. Furthermore, the ELDIT authoring tool is also an experiment to create an online learning system dedicated to university students to introduce structural linguistics and lexicography in a very practical way. It is intended as a cooperative approach – we might call it a 'controlled wiki approach' – where students complete online dictionary contents together with a tutor's constant feedback without the need for any massive external support.

1. The ELDIT dictionary: Lexicographic features

At the European Academy (EURAC) we have developed a computer-assisted language learning system called ELDIT (*Elektronisches Lern(er)wörterbuch Deutsch ITalienisch* 'Electronic learners' dictionary for German and Italian'). The main modules of the system are an electronic learners' dictionary, a learners' text corpus, a short grammar section and a module with quizzes. Each word in the whole system is annotated with a lemma and part-of-speech information, and is linked to the corresponding dictionary entry, which facilitates quick dictionary access for unknown words.

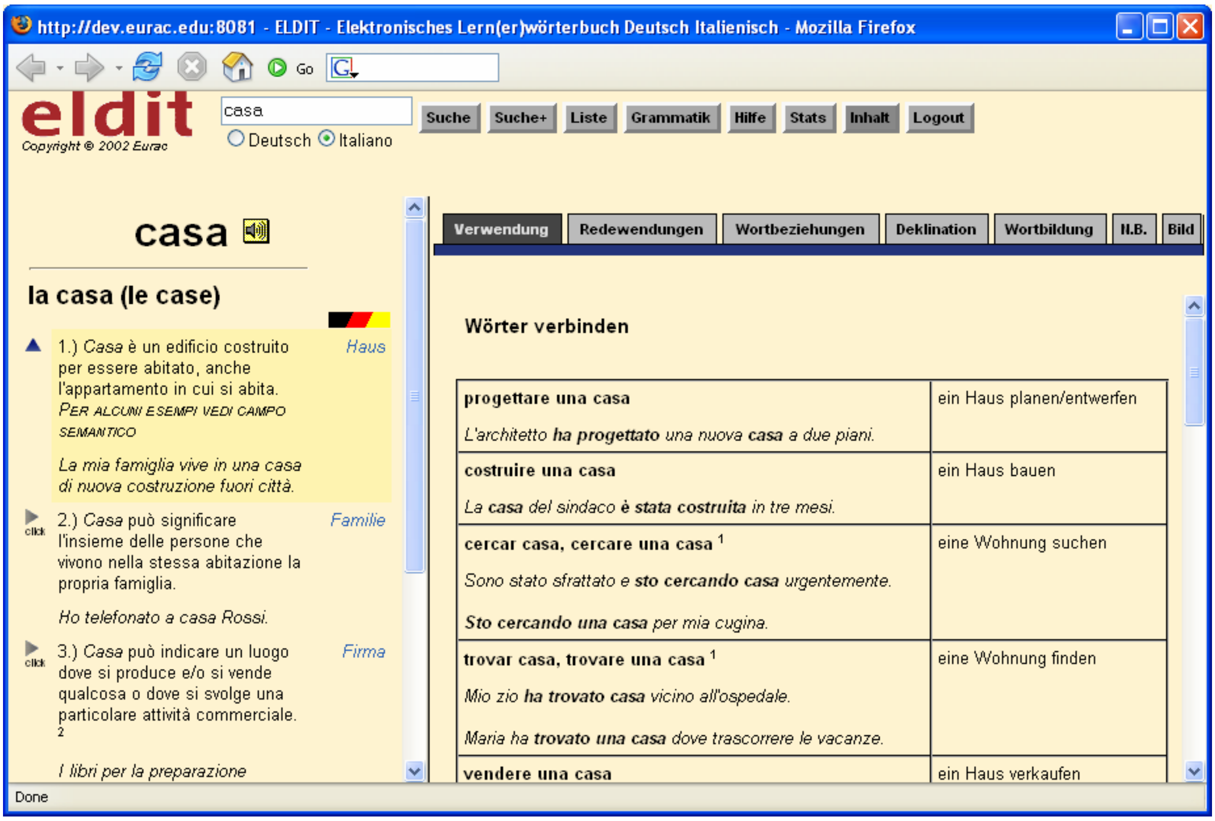
The innovative dictionary, which is the core part and the most advanced module of the system, is different from other lexicographic products regarding the following characteristics:

- ELDIT is a new type of dictionary, called 'cross-lingual' (Abel & Weber 2005). It is on the one hand designed as two separate monolingual dictionaries in that the meaning of each word is explained by a definition in the same language. This approach fulfils pedagogical demands which claim that it is better for the learner to remain in the target language. On the other hand the definitions are extended with translations, a typical element of bilingual dictionaries. This add-on fulfils the demands of learners who usually prefer bilingual dictionaries (Atkins & Knowles 1990; Nuccorini 1992). Moreover, the translation equivalent serves as an entry point to the corresponding part of the other module. In this way, with a simple mouse click, a user can easily switch between the two dictionary modules.
- ELDIT has been conceived for a precisely defined target group. Both the Italian and the German modules contain a basic vocabulary consisting of about 4,000 word entries. In this way we try to reach a target group which has been neglected in lexicography up until now, namely beginning to intermediate language learners (Waystage A1 – Threshold Level B1/B2). At the same time the dictionary has been

elaborated for the linguistic layman: this means that we have tried to avoid complicated linguistic expressions, metalanguage, abbreviations, and symbols in the interface as much as possible.

- ELDIT has been conceived as an electronic dictionary from the very beginning and tries to fully exploit the medium (Abel & Weber 2000); this distinguishes it from many other approaches which are conversions of paper dictionaries. The entries are organised in a very modular way and are highly interlinked (Gamper & Knapp 2003). Multimedia allows for the simultaneous use of different media such as text, images and sounds. Animations, movable elements and colours help to transmit complex linguistic information (Abel *et al*. 2003). The inclusion of computational linguistic tools allows for providing innovative features which are outstanding even within the most professional e-learning tools (Knapp *et al*. 2003; Knapp *et al*. 2004).

A dictionary entry is presented to the user in two frames, as seen in Figure 1. The left-hand frame shows the lemma of the word and a list of different word meanings, each of which is described by a definition, an example sentence, and one or more translation equivalents in the other language. The right-hand frame is organised modularly in several tabs and shows additional information such as word combinations, idioms, word fields, inflection, word families, linguistic difficulties, etc.



casa

la casa (le case)

1.) *Casa* è un edificio costruito per essere abitato, anche l'appartamento in cui si abita. *PER ALCUNI ESEMPI VEDI CAMPO SEMANTICO* — *Haus*

La mia famiglia vive in una casa di nuova costruzione fuori città.

2.) *Casa* può significare l'insieme delle persone che vivono nella stessa abitazione la propria famiglia. — *Familie*

Ho telefonato a casa Rossi.

3.) *Casa* può indicare un luogo dove si produce e/o si vende qualcosa o dove si svolge una particolare attività commerciale. 2 — *Firma*

I libri per la preparazione

Wörter verbinden

progettare una casa *L'architetto* ***ha progettato*** *una nuova* ***casa*** *a due piani.*	ein Haus planen/entwerfen
costruire una casa *La* ***casa*** *del sindaco* ***è stata costruita*** *in tre mesi.*	ein Haus bauen
cercar casa, cercare una casa [1] *Sono stato sfrattato e* ***sto cercando casa*** *urgentemente.* ***Sto cercando una casa*** *per mia cugina.*	eine Wohnung suchen
trovar casa, trovare una casa [1] *Mio zio* ***ha trovato casa*** *vicino all'ospedale.* *Maria ha* ***trovato una casa*** *dove trascorrere le vacanze.*	eine Wohnung finden
vendere una casa	ein Haus verkaufen

Figure 1: A dictionary entry in ELDIT: The Italian word *casa* 'house'

2. Problem description

The main purpose of online content is to increase the amount of knowledge on the Internet. It is also desirable to create resources with the possibility to expand all the contents at every point in time and by different contributors.

Initially, ELDIT needed only a quite simple DWS whose purpose was to create a fully-integrated instrument to support lexicographers. But, especially in our case of highly structured data, this kind of operation can be very complex for people with a basic IT knowledge and, therefore, reduce effectiveness and speed of updates.

The ELDIT collaborators were using a standard XML editor to accomplish this task, but after an in-depth analysis, we evaluated the possibility to generate an ad hoc interface to better support this important step of data creation, to reduce XML syntax errors, and to respond to another requirement that was expressed by universities and was related to the need to face the theoretical barrier within university education. Now we propose an electronic online tool with a double task. Firstly, the tool should make students more familiar with modern software and help avoiding a mainly theoretical introduction into linguistics and lexicography. Secondly, the prototype is studied by a team composed of linguists and computer scientists in order to create an ergonomic and adaptable interface that can easily decide which fields should be proposed to the student, or to the tutor, respectively.

Hence, the proposed tool should promote a close collaboration between students and teachers in the phase of learning and offer a technical solution towards a cooperative approach which splits the work not only between peers and tutors but also “between a human author and an intelligent authoring support tool, so that both human and AI agents are able to cooperate” (Brusilovsky *et al. forthcoming*).

3. Dictionary writing systems

A DWS can be described as “a piece of software for writing and producing a dictionary. It might include an editor, a database, a Web interface and various management tools (for allocating work etc.). It operates with a dictionary grammar, which specifies the structure of the dictionary” (Kilgarriff, page 7). Many DWSs and lexicographic tools exist. Each one fulfils the special needs of single dictionary projects. Some of them are free, some are commercial products, more or less adaptable and used in scientific contexts or by publishing houses, which build their own in-house systems. For an overview see for instance Joffe & De Schryver (2004).

Furthermore, a new trend can be noted, namely a collaborative approach whereby experts and non-experts – without any controlling authority – work together in order to create and improve new online lexicographic content using a freely available DWS as is for example the case for *Wiktionary*.

The need was felt at EURAC to create our own highly structured and flexible system that allows compiling new entries for the ELDIT dictionary and that interacts perfectly with the ELDIT system (containing very special features, modules and presentation possibilities from a linguistic as well as from a technical point of view, such as word field graphs, verb valency descriptions by means of movable images, inflectional tables, word families, cross-links, etc.). Now, some efforts have been made in order to render the tool easier to use and adaptable for other contexts and purposes.

4. Implementation issues

The XML (eXtensible Markup Language) was basically intended to manage interchanges between different sources. In our case the use of XML was necessary to provide a minimum of structure and to support data visualisation. The use of a DTD (document type definition) helps to standardise the input and to have basic checks on document consistency (Gamper & Knapp 2002; Gamper & Knapp 2003; Knapp *et al*. 2003).

Our first raw implementation of a basic DWS was a simple XML editor that was able to read a DTD and to provide a minimalist interface to create/remove/update elements and attributes into an XML document. Soon further modules were necessary to cross-link all the contexts and to index all the data that were contained within the 8,000 XML word entries. This kind of approach prevented the production of a more interactive dictionary and moving to a third version of the tool was necessary to better support the users. Hence, minimal requirements were established (interface independency from DTD, automatic approach to cross-linking and to indexing phases, immediate publication of the new contents), and so it became clear that we needed to modify the main engine to be a dictionary and at the same time to also give the opportunity to process data and to link them to all the rest of the context.

Access to XML documents was implemented using the *JDOM API*, but rapidly integrating cross-linking and indexing capabilities was our weak point in the workflow. For this reason, the idea to use a database came up (up to this point we had stored the XML documents on the disk). Only a few databases are able to provide the feature to index texts and XML documents in general. Two important ones are Tamino by *Software AG* and Oracle XML DB by *Oracle Corp*. These two databases presented the desired features which were the abilities to run an XPath query, and to run a full text search against all the data set. Oracle presented numerous advantages, particularly the ability to quickly deploy a data set and to have it indexed in a few minutes. Oracle also presented the possibility of supporting many more languages (23 languages at the moment) for full text search index capability, giving an assurance about the possibility to eventually extend the dictionary in the future.

The selection of this DB yielded the advantage of using SQL (Structured Query Language) instead of creating objects: SQL is more similar to human language for developers, so it is much easier to modify search patterns and to refine a search after inserting a new word; however, the XML objects are still within the design of the system, and are hidden by the SQL language of upper layers. The use of the Oracle XML DB required a conversion from the DTD to an XML Schema Definition (XSD), providing the advantage to manage the document structure more easily. The introduction of XSD led to the idea to use the XSD as a skeleton to fill a normal XML document, and to generate, based on XSLTs (Extensible Stylesheet Language Transformations), an HTML form to be filled. The XML form was enriched with a more suitable interface, and the basic idea was to maintain the data structure independent from its visualisation, as it had been possible with the XML editor.

The basic DTD/XSD template is used to produce the form, and the work flow engine solves problems such as paging and controls the entire data creation process. Previous conversion phases from the basic linguistic document to the fine-grained XML document is not performed at runtime but in background jobs without any user intervention; eventual problems are tracked by the logging system and submitted to the system administrator. Database triggers are also used to complete different views, to increase search speed, and to manage automatic indexing after data modification and insertion.

5. Data model

The ELDIT dictionary distinguishes four different classes of words: nouns, verbs, adjectives, and structure words. For each word class we define a corresponding DTD. Each dictionary entry represents a single word and is stored in a corresponding XML document. XML documents are organised into entities (table with some basic data and an additional XML content field) in the Oracle XML database.

A word in the ELDIT dictionary is described by a large amount of information of various types. For the purpose of language learning, a detailed representation of these data is required, since it is very important for the learner to recognise different parts of a word or a sentence. For example, the DTD of the noun class contains 39 elements. Some of the elements are compulsory, such as the element “lemma”, whereas others are optional, such as the element “derivation”. The data model provides the ability to manage compulsory and optional elements. If we look at the element “lemma” we can see that it is composed of just one ID and one #PCDATA (string) value:

```
<!ELEMENT lemma (#PCDATA)>
<!ATTLIST lemma id ID #IMPLIED >
```

Among other things we can decide to store a list of the most important derivations, distinguishing among prefix, base and suffix of a derivation. The corresponding part of the DTD is as follows:

```
<!ELEMENT derivation (prebasuf?,comment?,translation*)>
<!ATTLIST derivation id ID #IMPLIED >
<!ELEMENT pattern (rawData?,w*,nbs?)>
<!ATTLIST pattern id ID #IMPLIED restricted (yes|no) "no">
<!ELEMENT comment (rawData?,w*,nbs?)>
<!ATTLIST comment id ID #IMPLIED position (before|after) "before">
<!ELEMENT translation (rawData?,w*,nbs?)>
<!ATTLIST translation id ID #IMPLIED>
```

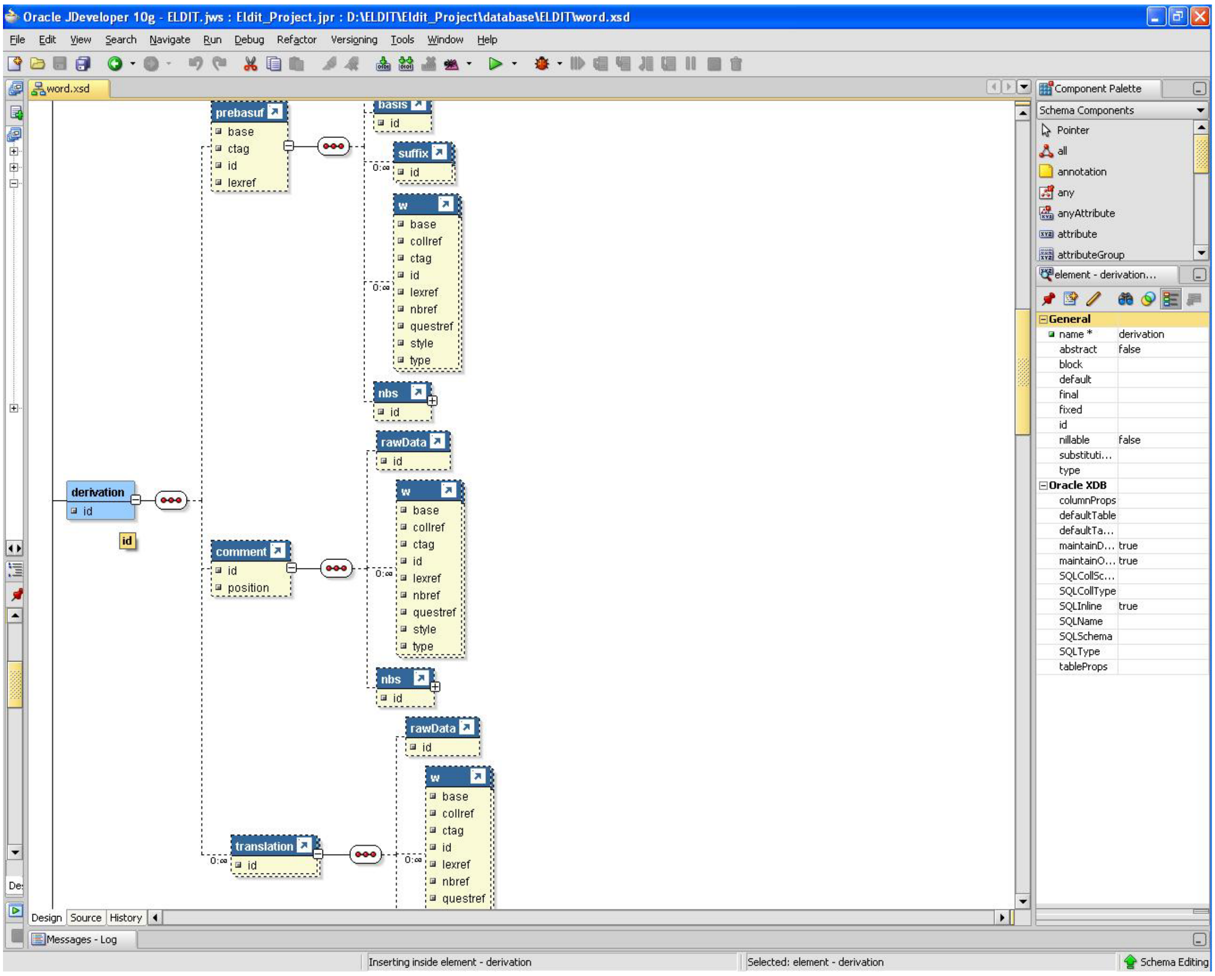
The element derivation consists of three sub-elements: a `prebasuf` (= prefix-basis-suffix), a comment, and possibly some translations. The `prebasuf` is subdivided into article, prefixes, basis and suffixes. Comments and translations are composed of a sequence of at least one token each. For example, the word *Haus* has a derivation *Behausung*, which can be split up into the prefix *Be*, the basis *haus*, and the suffix *ung*. The article is *die*, and the translation is *la dimora*. The corresponding part of the XML file is shown below:

```
<derivation id="de.n.haus.1.word0.noun0.derivation2">
   <prebasuf base="Behausung" ctag="N" lexref="de.n.hochhaus.1.word0.noun0.derivation2.
   prebasuf0" id="de.n.haus.1.word0.noun0.derivation2.prebasuf0">
      <article ctag="S" base="d" lexref="de.g.artikel.1.LexGram0.features0.item0.gr_lem
      ma0.w0" id="de.n.haus.1.word0.noun0.derivation2.prebasuf0.article0">die</article>
      <praefix id="de.n.haus.1.word0.noun0.derivation2.prebasuf0.praefix0">Be</praefix>
      <basis id="de.n.haus.1.word0.noun0.derivation2.prebasuf0.basis0">haus</basis>
      <suffix id="de.n.haus.1.word0.noun0.derivation2.prebasuf0.suffix0">ung</suffix>
   </prebasuf>
   <translation id="de.n.haus.1.word0.noun0.derivation2.translation0">
      <w ctag="S" base="il" lexref="it.g.articolo.1.LexGram0.features0.item0.gr_lemma0.
      w0" id="de.n.haus.1.word0.noun0.derivation2.translation0.w0">la</w>
      <w type="space" ctag="PON:*:*:*:*:*" base="" lexref="SSS" id="de.n.haus.1.word0.
      noun0.derivation2.translation0.w1"></w>
      <w ctag="N" base="dimora" lexref="KKK" id="de.n.haus.1.word0.noun0.derivation2.
      translation0.w2">dimora</w>
   </translation>
</derivation>
```

A hierarchical view of the same DTD part may be seen in Figure 2, which indicates the complexity of this specific element. The element <w> is the lowest level element which contains the raw data (#PCDATA). Almost all other elements (definition, translation, example, footnote, explanation, grammatical indication, etc.) are described by this element. The corresponding part of the DTD is as follows:

```
<!ELEMENT w (#PCDATA)>
<!ATTLIST w id   ID #IMPLIED
                 type CDATA #IMPLIED
                 style CDATA #IMPLIED
                 nbref CDATA #IMPLIED
                 questref CDATA #IMPLIED
                 base CDATA #IMPLIED
                 ctag CDATA #IMPLIED
                 lexref CDATA #IMPLIED
                 collref CDATA #IMPLIED>
```

The <w>-element defines several attributes, which provide additional information. The attribute type specifies whether the enclosed data are content data or a remark. The attribute style indicates which data have to be emphasized. The attribute “lexref” contains a reference to another dictionary entry. In this way we can link the words in definitions, sample sentences, collocations, etc. to the corresponding dictionary entries. The attribute “nbref” contains a reference to a footnote within the same XML document.



Oracle JDeveloper 10g - ELDIT.jws : Eldit_Project.jpr : D:\ELDIT\Eldit_Project\database\ELDIT\word.xsd

element - derivation...

General	
name *	derivation
abstract	false
block	
default	
final	
fixed	
id	
nillable	false
substituti...	
type	
Oracle XDB	
columnProps	
defaultTable	
defaultTa...	
maintainD...	true
maintainO...	true
SQLCollSc...	
SQLCollType	
SQLInline	true
SQLName	
SQLSchema	
SQLType	
tableProps	

Inserting inside element - derivation

Selected: element - derivation

Figure 2: Hierarchical view of the word-DTD: The element “derivation” is an optional child of the element “noun”

6. Towards an online dictionary writing system: A cooperative approach

Three actors are mainly involved in the whole process: a student, a teacher/tutor and the program itself in the sense that it helps automatically integrating content using the ELDIT system. Figure 3 shows the workflow subdivided in different steps.

First, the student creates a new dictionary entry for a lemma, elaborating the data and inserting them into a form, presented to him/her on the basis of his/her skills and experience. Afterwards, the data are sent to the ELDIT database and are enriched with new content, for instance inflectional information and word families. The data are also pre-indexed and pre-linked to the corresponding dictionary entries within the system. In case the system is not able to integrate some data the file is sent back to the student and he/she has to complete it (e.g.

word disambiguation when there are two different lemmas for an inflected word form) or create material from scratch (e.g. word families).

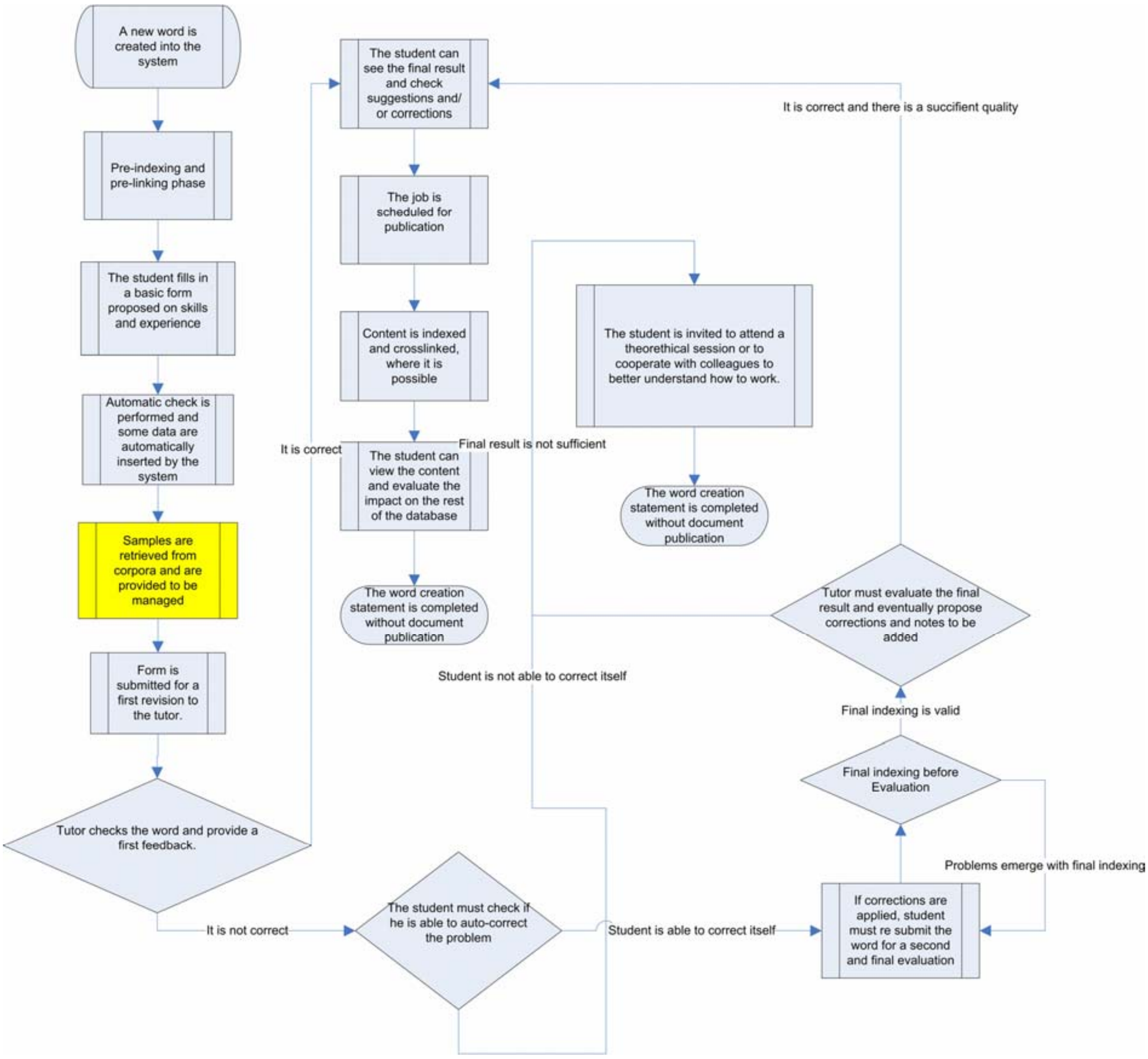

A new word is created into the system
Pre-indexing and pre-linking phase
The student fills in a basic form proposed on skills and experience
Automatic check is performed and some data are automatically inserted by the system
Samples are retrieved from corpora and are provided to be managed
Form is submitted for a first revision to the tutor.
Tutor checks the word and provide a first feedback.
It is correct
It is not correct
The student must check if he is able to auto-correct the problem
Student is not able to correct itself
Student is able to correct itself
The student can see the final result and check suggestions and/ or corrections
The job is scheduled for publication
Content is indexed and crosslinked, where it is possible
The student can view the content and evaluate the impact on the rest of the database
Final result is not sufficient
The word creation statement is completed without document publication
The student is invited to attend a theorethical session or to cooperate with colleagues to better understand how to work.
The word creation statement is completed without document publication
It is correct and there is a succifient quality
Tutor must evaluate the final result and eventually propose corrections and notes to be added
Final indexing is valid
Final indexing before Evaluation
Problems emerge with final indexing
If corrections are applied, student must re submit the word for a second and final evaluation

Figure 3: Workflow of the ELDIT online dictionary writing system

Between this and the next step an additional one is planned: Always starting from the same authoring tool, data extraction from corpora should be possible. Therefore, students can use a concordance tool which helps to check existing word combinations, or to find new ones which have to be inserted into the dictionary. Furthermore, on the basis of a given word combination pattern and a special algorithm, the system will be able to search the corpus for lexicographic examples, which can be accepted as they are, modified or rejected.

After these steps the form is submitted for a first revision to the tutor who checks the data and provides feedback. If the dictionary entry isn't complete, or if one or more errors have to be corrected, the form is re-sent to the student for a revision. Now the student has to decide if he/she is able to solve the problem(s) by him/herself or not. In the first case, he/she does the corrections and the form is submitted to the tutor for a second evaluation. In the second case, the student can call for help (it has still to be decided which kind of support this will be, e.g. a theoretical session, an online tutorial, or …). Only after this additional step the student can do the necessary corrections and submit the form to the tutor. Theoretically the re-submitting and controlling process between student and tutor can be repeated ad infinitum. As soon as the tutor is satisfied with the outcome, he can evaluate the result and give a final comment or a mark. The form with the final comments is sent back to the student.

Then the data undergo a final indexing and linking process; in case the system detects ambiguities (e.g. word classes, or …), the student has to intervene once more. Finally, the form is scheduled for publication. The student can view the content and evaluate the impact on the rest of the database. The word creation process is completed as soon as the document is published.

7. Conclusions and outlook

What was presented in this paper is a draft idea and implementation of a new online tool specific to the ELDIT database, which aims to permit users to have a more comfortable interface to insert data or to correct data if it is needed. The proposed interface is user-friendly and widely available thanks to modern Web client technologies. The adaptive interface also gives the possibility of having a user-driven system, and of mitigating the sense of frustration of some users who find some tools to be too complicated.

As a next step, our goal is to have the ability to link the authoring tool with external resources, especially with corpora, to retrieve authentic samples that can be used by the students to better clarify the use of a given word, and to have an immediate suggestion and feedback from other online resources. We hope this task will also give the students insight into the use of corpora, and show that online resources are valuable tools if used together to refine the final result.

Finally, we propose to use ELDIT as a sample of a technically well-structured system, which can be re-adapted to the needs of other online dictionary projects and used for a faster deployment of new bilingual dictionaries with minimal effort.

References

Abel, A., J. Gamper, J. Knapp & V. Weber. 2003. Describing Verb Valency in an Electronic Learner’s Dictionary: Linguistic and Technical Implications. In Lassner, D. & C. McNaught (eds.). 2003. *Proceedings of ED-MEDIA 2003, World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, USA*: 1202–1209.

Abel, A. & V. Weber. 2000. ELDIT – A Prototype of an Innovative Dictionary. In Heid, U., S. Evert, E. Lehmann & C. Rohrer (eds.). 2000. *Proceedings of the Ninth EURALEX International Congress, EURALEX 2000, Stuttgart, Germany, August 8th-12th, 2000*: 807–818. Stuttgart: Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Abel, A. & V. Weber. 2005. ELDIT – Electronic Learner’s Dictionary of German and Italian: Semibilingual, Bilingualised or a Totally New Type? In Gottlieb, H., J.E. Mogensen & A. Zettersten (eds.). 2005. *Symposium on Lexicography XI, Proceedings of the Eleventh International Symposium on Lexicography, May 2-4, 2002, at the University of Copenhagen* (Lexicographica Series Maior 115): 73–84. Tübingen: Max Niemeyer.

Atkins, B.T.S. & F.E. Knowles. 1990. Interim report on the EURALEX/AILA research project into dictionary use. In Magay, T. & J. Zigány (eds.). 1990. *BudaLEX ’88 Proceedings, Papers from the EURALEX Third International Congress*: 381–392. Budapest: Akadémiai Kiadó.

Brusilovsky, P., J. Knapp & J. Gamper. *forthcoming*. Supporting teachers as content authors in intelligent educational systems. *International Journal of Knowledge and Learning*.

ELDIT. Available from `http://dev.eurac.edu:8081/MakeEldit1/Eldit.html`

Gamper, J. & J. Knapp. 2002. XML for an electronic learners’ dictionary. In *Proceedings of the IADIS International Conference, WWW/Internet 2002, Lisbon, Portugal*: 427–434.

Gamper, J. & J. Knapp. 2003. A data model and its implementation for a Web-based language learning system. In *Proceedings of the 12th International World Wide Web Conference, WWW 2003*: 217–225.

JDOM API. Available from `http://www.jdom.org/`

Joffe, D. & G-M de Schryver. 2004. TshwaneLex – A State-of-the-Art Dictionary Compilation Program. In Williams, G. & S. Vessier (eds.). 2004. *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004, Lorient, France, July 6-10, 2004*: 99–104. Lorient: Faculté des Lettres et des Sciences Humaines, Université de Bretagne Sud.

Knapp, J., P. ten Hacken & S. Pedrazzini. 2003. ELDIT and WordManager, a powerful partnership. In Lassner, D. & C. McNaught (eds.). 2003. *Proceedings of ED-MEDIA 2003, World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, USA*.

Knapp, J., J. Gamper & P. Brusilovsky. 2004. Reuse of Lexicographic Examples in a Web-based Learners' Dictionary. In *Proceedings of e-Learn 2004*.

Nuccorini, S. 1992. Monitoring Dictionary Use. In Tommola, H., K. Varantola, T. Salmi-Tolonen & J. Schopp. (eds.). 1992. *EURALEX '92 PROCEEDINGS I-II, Papers submitted to the 5th EURALEX International Congress on Lexicography in Tampere, Finland* (Studia Translatologica A/2): 89–102. Tampere: Department of Translation Studies, University of Tampere.

Wiktionary. Available from `http://en.wiktionary.org/wiki/Main_Page`

Software demo: The terminographic processor MyTerMS

Igor Kudashev & Irina Kudasheva

Palmenia Centre for Continuing Education, University of Helsinki, Kouvola, Finland
E-mail: {igor.kudashev,irina.kudasheva}@helsinki.fi
Web: http://www.helsinki.fi/palmenia/english/

Abstract The MyTerMS terminographic processor was designed in 2004 for the Finnish–Russian Forestry Dictionary Project (2003–2006), financed by the E.U. and the State Provincial Office of Southern Finland. Later it was adapted for the Finnish–Russian Dictionary of Idioms and for the Common Language project aimed at the compilation of an English–Finnish–Russian glossary of the E.U. Neighbourhood Programme terminology. In this paper we will discuss the Forestry Dictionary version of the program as it is the most advanced one.

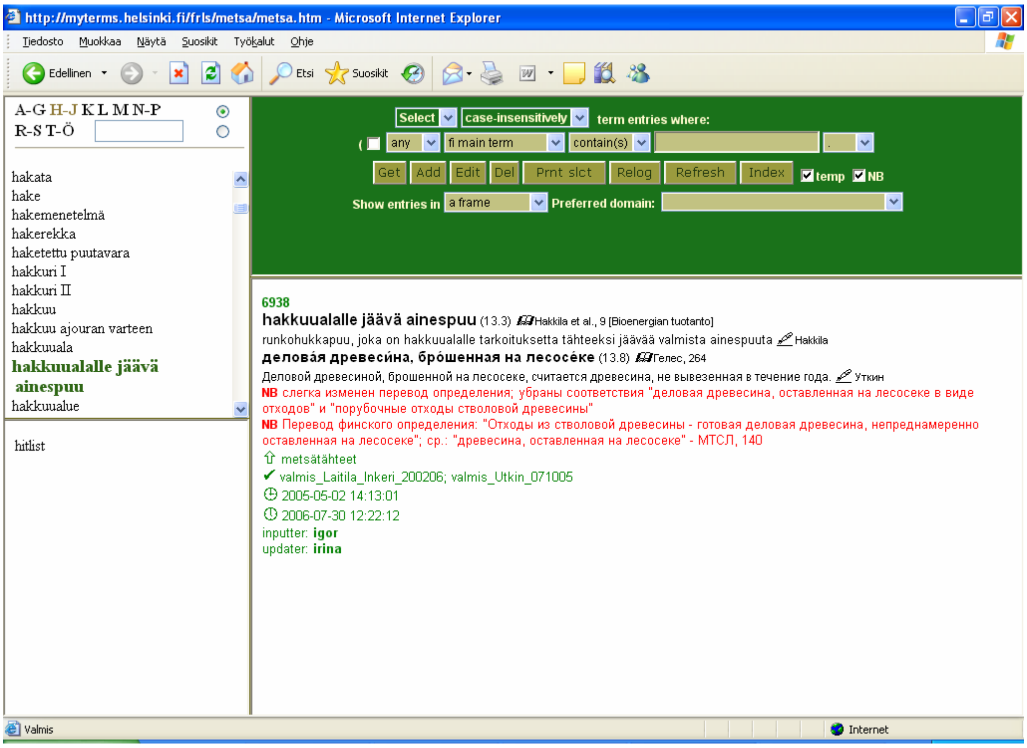
The name of the program comes from the words 'My Terminology Management System' although it is only one of many possible interpretations. MyTerMS was designed as a terminology management system but its use in the compilation of a dictionary of idioms demonstrates that its application is not restricted to terminological products. MyTerMS performs all basic operations which can be expected from dictionary management software such as adding, editing, searching, browsing, printing and deleting the entries. In addition, it automates many operations and helps ensure the integrity of the data and the correctness of the input.

MyTerMS is a Web interface to the underlying dictionary databases stored in the MySQL 5 database management program. The functions of the HTTP server are performed by Apache 2 with the ModPerl 2 module. Programming is done with Perl and DHTML. Internet Explorer 5.5+ is required for the program to work correctly.

The first window the user encounters when he/she starts the program is the login screen. Here the user has to specify the database, the login and the password. These are used to identify the user in the database management program. Global authorisation is not used because the full-fledged version of the program is accessible only from specified IP-addresses. If the user has mistyped any of the fields or he/she wishes to log in under another name or into another database, he/she can press the Relog button.

After the user has logged in, the list of lemmata is loaded into the upper left-hand frame of the screen. Since the list may contain thousands of lemmata, it is divided into several parts. The user can switch between the portions of the lemmata list by clicking on the corresponding letters or ranges of letters. Individual entries are selected by clicking on the entry head. By default the entries are displayed in the lower right-hand frame but they can also be displayed in a new window.

The layout of the entries is as close to final as possible except for the presence of some temporary and administrative data which is visually separated from the final data with colour. Temporary and administrative data can be switched on and off by ticking the "temp" and the "NB" checkboxes – see Figures 1 versus 2. The entries are segmented into data categories and the articles are formed 'on the fly' with the help of scripts and cascading style sheets. This means that the layout and the structure of the entries can be easily altered whenever necessary. Searching possibilities of the program are rather extensive. The search can be performed in any of the data categories present in the database. AND/OR operators can be used to combine up to four search conditions. If both AND and OR operators are used, the precedence of operations can be specified with the help of brackets.

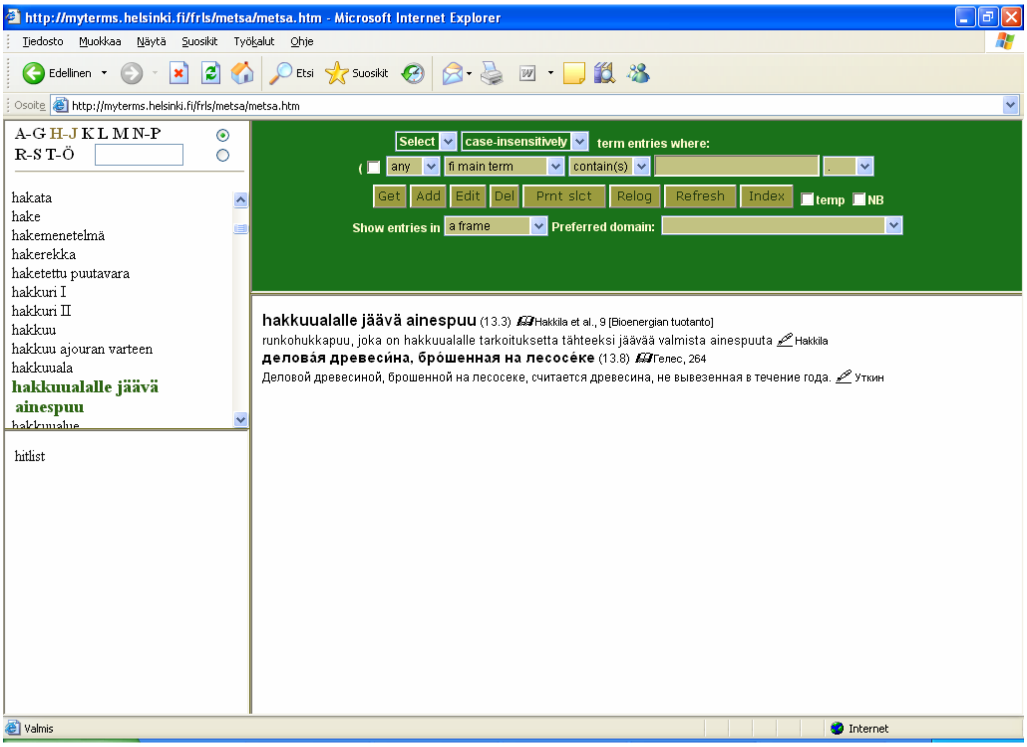


☑ temp ☑ NB

Show entries in: a frame
Preferred domain:

6938
hakkuualalle jäävä ainespuu (13.3) Hakkila et al., 9 [Bioenergian tuotanto]
runkohukkapuu, joka on hakkuualalle tarkoituksetta tähteeksi jäävää valmista ainespuuta Hakkila
делова́я древеси́на, бро́шенная на лесосе́ке (13.8) Гелес, 264
Деловой древесиной, брошенной на лесосеке, считается древесина, не вывезенная в течение года. Уткин
NB слегка изменен перевод определения; убраны соответствия "деловая древесина, оставленная на лесосеке в виде отходов" и "порубочные отходы стволовой древесины"
NB Перевод финского определения: "Отходы из стволовой древесины - готовая деловая древесина, непреднамеренно оставленная на лесосеке"; ср.: "древесина, оставленная на лесосеке" - МТСЛ, 140
⇧ metsätähteet
✓ valmis_Laitila_Inkeri_200206; valmis_Utkin_071005
2005-05-02 14:13:01
2006-07-30 12:22:12
inputter: **igor**
updater: **irina**

Figure 1: Temporary (red) and administrative (green) data is switched on



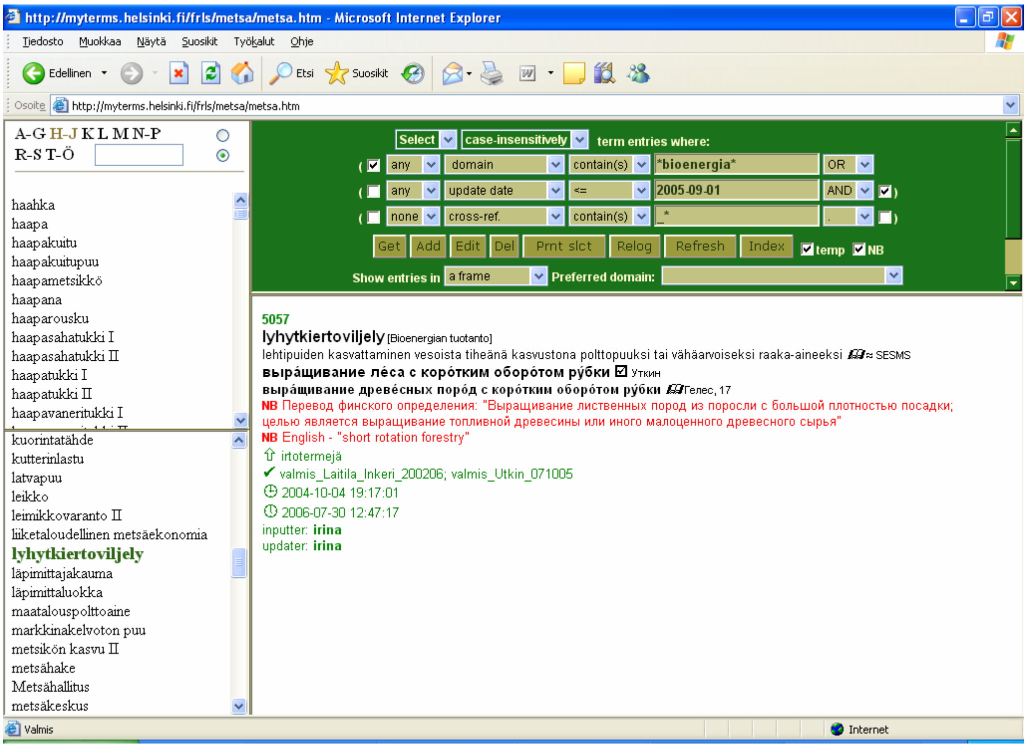
☐ temp ☐ NB

Show entries in: a frame
Preferred domain:

hakkuualalle jäävä ainespuu (13.3) Hakkila et al., 9 [Bioenergian tuotanto]
runkohukkapuu, joka on hakkuualalle tarkoituksetta tähteeksi jäävää valmista ainespuuta Hakkila
делова́я древеси́на, бро́шенная на лесосе́ке (13.8) Гелес, 264
Деловой древесиной, брошенной на лесосеке, считается древесина, не вывезенная в течение года. Уткин

Figure 2: Final layout: Temporary and administrative data are switched off

All operators used in the MySQL database management software can be used in the search, including exact match, wildcard search, numerical comparison and regular expressions. A more familiar asterisk is used to substitute the SQL percentage sign as a wildcard symbol. The search may be case-sensitive or case-insensitive. Depending on the user's choice the program either displays the entry heads of the articles which correspond to the criteria, in the lower left-hand frame, see Figure 3, or simply counts them. If the number of hits is very large which implies a possibility of a mistake in the search conditions, the program asks the user for confirmation before it displays the results.


http://myterms.helsinki.fi/frls/metsa/metsa.htm - Microsoft Internet Explorer
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Edellinen
Etsi
Suosikit
Osoite
Get
Add
Edit
Del
Prnt slct
Relog
Refresh
Index
Valmis
Internet

http://myterms.helsinki.fi/frls/metsa/metsa.htm

A-G H-J K L M N-P ○
R-S T-Ö ●

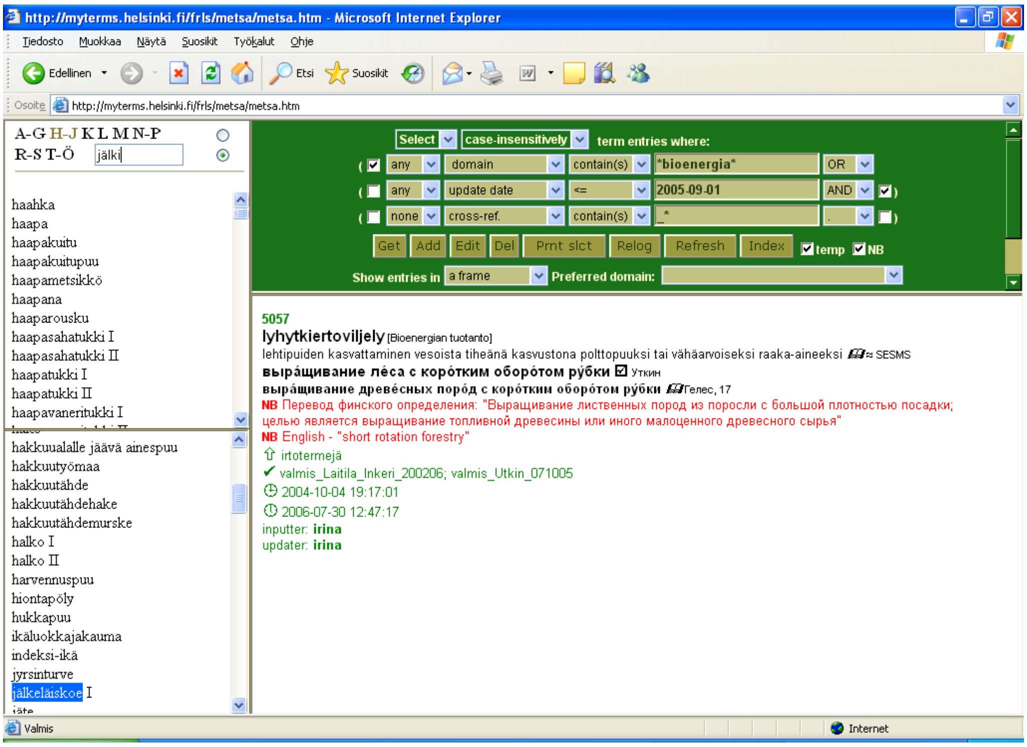
Select case-insensitively term entries where:
(☑ any domain contain(s) *bioenergia* OR
(☐ any update date <= 2005-09-01 AND ☑)
(☐ none cross-ref. contain(s) _* . ☐)
☑ temp ☑ NB
Show entries in a frame Preferred domain:

haahka
haapa
haapakuitu
haapakuitupuu
haapametsikkö
haapana
haaparousku
haapasahatukki I
haapasahatukki II
haapatukki I
haapatukki II
haapavaneritukki I

kuorintatähde
kutterinlastu
latvapuu
leikko
leimikkovaranto II
liiketaloudellinen metsäekonomia
lyhytkiertoviljely
läpimittajakauma
läpimittaluokka
maatalouspolttoaine
markkinakelvoton puu
metsikön kasvu II
metsähake
Metsähallitus
metsäkeskus

5057
lyhytkiertoviljely [Bioenergian tuotanto]
lehtipuiden kasvattaminen vesoista tiheänä kasvustona polttopuuksi tai vähäarvoiseksi raaka-aineeksi ≈ SESMS
вырáщивание лéса с корóтким оборóтом рýбки ☑ Уткин
вырáщивание древéсных порóд с корóтким оборóтом рýбки Гелес, 17
NB Перевод финского определения: "Выращивание лиственных пород из поросли с большой плотностью посадки; целью является выращивание топливной древесины или иного малоценного древесного сырья"
NB English - "short rotation forestry"
⇧ irtotermejä
✓ valmis_Laitila_Inkeri_200206; valmis_Utkin_071005
2004-10-04 19:17:01
2006-07-30 12:47:17
inputter: **irina**
updater: **irina**

Figure 3: Results of a multi-conditional search are displayed in the ‘hitlist’


http://myterms.helsinki.fi/frls/metsa/metsa.htm - Microsoft Internet Explorer
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Edellinen
Etsi
Suosikit
Osoite
Get
Add
Edit
Del
Prnt slct
Relog
Refresh
Index
Valmis
Internet

A-G H-J K L M N-P ○
R-S T-Ö jälki ●

hakkuualalle jäävä ainespuu
hakkuutyömaa
hakkuutähde
hakkuutähdehake
hakkuutähdemurske
halko I
halko II
harvennuspuu
hiontapöly
hukkapuu
ikäluokkajakauma
indeksi-ikä
jyrsinturve
jälkeläiskoe I

http://myterms.helsinki.fi/frls/metsa/metsa.htm

Select case-insensitively term entries where:
(☑ any domain contain(s) *bioenergia* OR
(☐ any update date <= 2005-09-01 AND ☑)
(☐ none cross-ref. contain(s) _* . ☐)
☑ temp ☑ NB
Show entries in a frame Preferred domain:

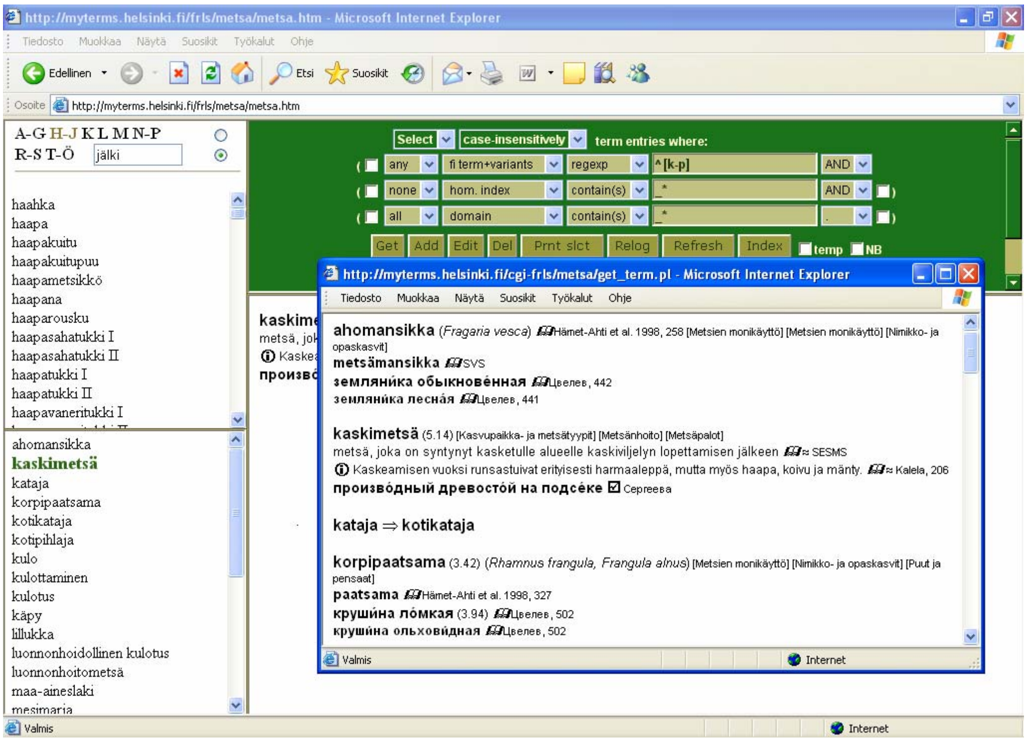
haahka
haapa
haapakuitu
haapakuitupuu
haapametsikkö
haapana
haaparousku
haapasahatukki I
haapasahatukki II
haapatukki I
haapatukki II
haapavaneritukki I

5057
lyhytkiertoviljely [Bioenergian tuotanto]
lehtipuiden kasvattaminen vesoista tiheänä kasvustona polttopuuksi tai vähäarvoiseksi raaka-aineeksi ≈ SESMS
вырáщивание лéса с корóтким оборóтом рýбки ☑ Уткин
вырáщивание древéсных порóд с корóтким оборóтом рýбки Гелес, 17
NB Перевод финского определения: "Выращивание лиственных пород из поросли с большой плотностью посадки; целью является выращивание топливной древесины или иного малоценного древесного сырья"
NB English - "short rotation forestry"
⇧ irtotermejä
✓ valmis_Laitila_Inkeri_200206; valmis_Utkin_071005
2004-10-04 19:17:01
2006-07-30 12:47:17
inputter: **irina**
updater: **irina**

Figure 4: The ‘quick search’ scrolls into view and highlights the closest match

The entries in the ‘hitlist’ can be browsed in the same way as the entries in the main list. Navigation through the main list and the hitlist is facilitated with a ‘quick search’ when the user types only several initial letters of the lemma he/she is searching. The first exact match or the closest match found is scrolled into view and highlighted, see Figure 4.

The list of entries found in the hitlist can be displayed in a separate window from where it can be printed or saved as an HTML file in UTF-8 encoding for further processing, see Figure 5. Since the language of most data categories is specified, the text of the dictionary can be spellchecked in applications like *Microsoft* Word. Hyphenation works, too, which is important for a better and space-saving layout.


http://myterms.helsinki.fi/frls/metsa/metsa.htm - Microsoft Internet Explorer
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Edellinen
Etsi
Suosikit
Osoite http://myterms.helsinki.fi/frls/metsa/metsa.htm
A-G H-J K L M N-P ○
R-S T-Ö jälki ●
haahka
haapa
haapakuitu
haapakuitupuu
haapametsikkö
haapana
haaparousku
haapasahatukki I
haapasahatukki II
haapatukki I
haapatukki II
haapavaneritukki I
ahomansikka
kaskimetsä
kataja
korpipaatsama
kotikataja
kotipihlaja
kulo
kulottaminen
kulotus
käpy
lillukka
luonnonhoidollinen kulotus
luonnonhoitometsä
maa-aineslaki
mesimarja
Select case-insensitively term entries where:
(□ any fi term+variants regexp ^[k-p] AND
(□ none hom. index contain(s) _* AND □)
(□ all domain contain(s) _* . □)
Get Add Edit Del Prnt slct Relog Refresh Index □ temp □ NB
http://myterms.helsinki.fi/cgi-frls/metsa/get_term.pl - Microsoft Internet Explorer
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Valmis
Internet

ahomansikka (*Fragaria vesca*) Hämet-Ahti et al. 1998, 258 [Metsien monikäyttö] [Metsien monikäyttö] [Nimikko- ja opaskasvit]
metsämansikka SVS
земляни́ка обыкнове́нная Цвелев, 442
земляни́ка лесна́я Цвелев, 441

kaskimetsä (5.14) [Kasvupaikka- ja metsätyypit] [Metsänhoito] [Metsäpalot]
metsä, joka on syntynyt kasketulle alueelle kaskiviljelyn lopettamisen jälkeen ≈ SESMS
ⓘ Kaskeamisen vuoksi runsastuivat erityisesti harmaaleppä, mutta myös haapa, koivu ja mänty. ≈ Kalela, 206
произво́дный древосто́й на подсе́ке ☑ Сергеева

kataja ⇒ kotikataja

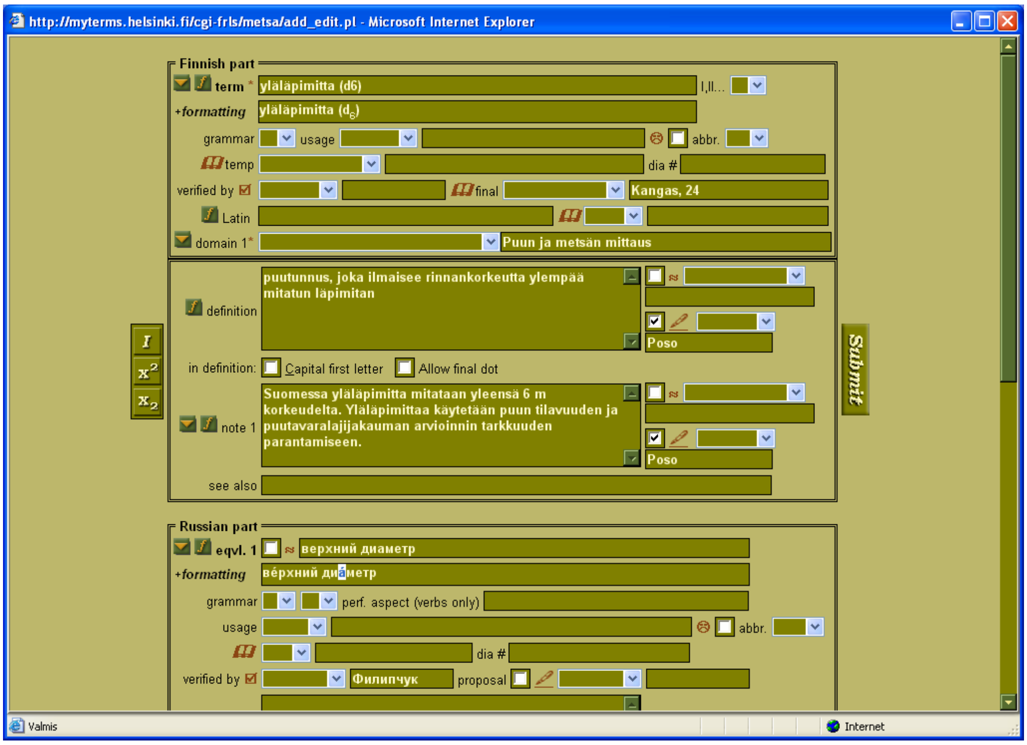
korpipaatsama (3.42) (*Rhamnus frangula, Frangula alnus*) [Metsien monikäyttö] [Nimikko- ja opaskasvit] [Puut ja pensaat]
paatsama Hämet-Ahti et al. 1998, 327
круши́на ло́мкая (3.94) Цвелев, 502
круши́на ольхови́дная Цвелев, 502

Figure 5: The ‘hitlist’ entries are displayed in a window from where they can be saved

Entries can be added and edited with an HTML form. Two fields are mandatory: the lemma and the domain to which it belongs. Additional fields and groups of fields can be revealed by pressing the “Arrow down” button. For example, it is possible to add up to six variants / synonyms of the lemma and as many target-language equivalents.

Some fields are a combination of a pick-up menu and a free-style field. For example, in the usage field the user can choose a predefined value (rare, slang, etc.) or write a comment of his/her own (e.g. “used in …”). Similarly, predefined abbreviations of the written sources can be supplemented with page numbers.

Normally, the layout of each data category is defined by the style sheet. However, at some stage we found it necessary to manually add layout to some fields. For example, sometimes it is necessary to stress a segment of a note with italics or to use upper or lower case. Some equivalents in our dictionary had to be interrupted with short additional explanations in italics. Besides, we used a special font with stressed vocals for the Russian equivalents. If tags and additional strings were used within the data categories, the search by means of the database management system would have become problematic. To avoid this, fields with formatting and supplements are duplicated in the database in a basic, plain text form as well. To add formatting to a field, the user has to fill in the plain text field first and then to press a button which copies the contents into the formattable field. In the formattable field, the user can add comments to the string or add formatting, for example using the Word-like scrollable instrument pane on his/her left-hand side, as shown in Figure 6. The formatted string is displayed in the entry but there is also a plain text version in the database which is used for the purposes of searching and building an index. The same method can be used for adding grammatical labels to term elements within the term like in "**means** *pl*. **of payment**".



http://myterms.helsinki.fi/cgi-frls/metsa/add_edit.pl - Microsoft Internet Explorer

Finnish part

term * yläläpimitta (d6)

+formatting yläläpimitta (d_6)

verified by ☑ final Kangas, 24

domain 1* Puun ja metsän mittaus

definition: puutunnus, joka ilmaisee rinnankorkeutta ylempää mitatun läpimitan — ☑ Poso

in definition: □ Capital first letter □ Allow final dot

note 1: Suomessa yläläpimitta mitataan yleensä 6 m korkeudelta. Yläläpimittaa käytetään puun tilavuuden ja puutavaralajijakauman arvioinnin tarkkuuden parantamiseen. — ☑ Poso

see also

Russian part

eqvl. 1 □ верхний диаметр

+formatting вéрхний диáметр

verified by ☑ Филипчук proposal □

Figure 6: Stress marks, subscripts, etc. can be added to formattable fields

The program adds all special signs and brackets only when forming an entry. This prevents input errors, makes terminologists' life easier and allows altering one special sign or type of brackets into another if necessary without making changes to the contents of the database.

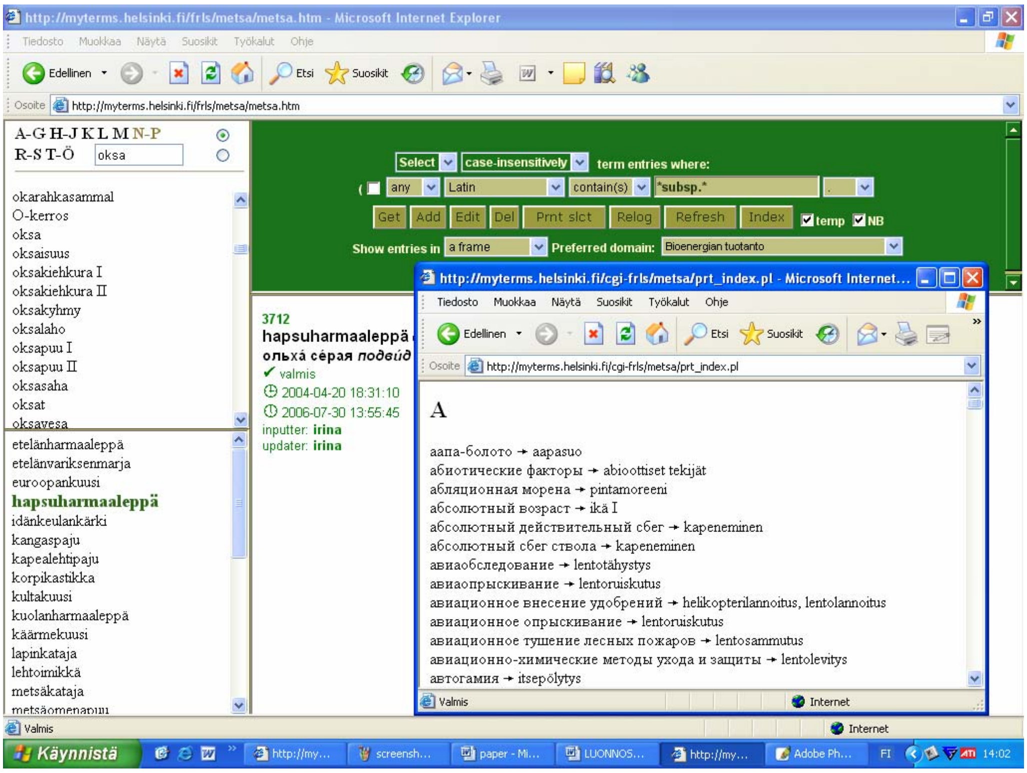
The program makes the first letter lower-case and deletes the final dot in the definition field and does vice versa in the note fields. However, it is possible to make exceptions when a definition starts with a capital letter or ends in a dot according to grammatical rules.

The program prevents duplicate entries; a correct homonym index has to be specified for homonyms. Cross-references are generated, edited and deleted fully automatically when the user edits the main entry. Manual editing of cross-reference articles is not permitted. The domain is normally not specified in cross-reference articles except for the cases when reference articles contain homonyms.

The program checks that mutually exclusive fields are not filled in simultaneously. For example, a term proposed by an expert cannot be marked as non-recommended or have a written source. Daughter fields cannot be filled in when the mother field is not filled in. For example, the source of the definition cannot be specified if the definition field itself is empty. All these checks are performed when the user submits the form.

If the user is going to add a number of lemmata belonging to the same domain, he/she can specify it in the 'Preferred domain' menu. The domain is automatically selected in the add/edit form.

The program automatically builds an alphabetical index of the target-language lemmata, see Figure 7. Correct alphabetisation of the source and target-language lemmata required several algorithms. Problems included non-alphabetical symbols and space marks, small and capital letters, *v* and *w* which are treated as equal letters in the Finnish language, etc.


http://myterms.helsinki.fi/frls/metsa/metsa.htm - Microsoft Internet Explorer
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Edellinen Etsi Suosikit
Osoite http://myterms.helsinki.fi/frls/metsa/metsa.htm
A-G H-J K L M N-P
R-S T-Ö
Select
case-insensitively
term entries where:
Get Add Edit Del Prnt slct Relog Refresh Index
Show entries in
Preferred domain:
http://myterms.helsinki.fi/cgi-frls/metsa/prt_index.pl - Microsoft Internet…
Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje
Edellinen Etsi Suosikit
Osoite http://myterms.helsinki.fi/cgi-frls/metsa/prt_index.pl
Valmis
Internet
Valmis
Internet
Käynnistä
http://my…
screensh…
paper - Mi…
LUONNOS…
http://my…
Adobe Ph…
FI
14:02

● A-G H-J K L M N-P
○ R-S T-Ö oksa

okarahkasammal
O-kerros
oksa
oksaisuus
oksakiehkura I
oksakiehkura II
oksakyhmy
oksalaho
oksapuu I
oksapuu II
oksasaha
oksat
oksavesa

etelänharmaaleppä
etelänvariksenmarja
euroopankuusi
hapsuharmaaleppä
idänkeulankärki
kangaspaju
kapealehtipaju
korpikastikka
kultakuusi
kuolanharmaaleppä
käärmekuusi
lapinkataja
lehtoimikkä
metsäkataja
metsäomenapuu

Select case-insensitively term entries where:
(□ any Latin contain(s) *subsp.*
☑ temp ☑ NB
Show entries in a frame Preferred domain: Bioenergian tuotanto

3712
hapsuharmaaleppä
ольха́ се́рая *подви́д*
✓ valmis
2004-04-20 18:31:10
2006-07-30 13:55:45
inputter: **irina**
updater: **irina**

A

аапа-болото → aapasuo
абиотические факторы → abioottiset tekijät
абляционная морена → pintamoreeni
абсолютный возраст → ikä I
абсолютный действительный сбег → kapeneminen
абсолютный сбег ствола → kapeneminen
авиаобследование → lentotähystys
авиаопрыскивание → lentoruiskutus
авиационное внесение удобрений → helikopterilannoitus, lentolannoitus
авиационное опрыскивание → lentoruiskutus
авиационное тушение лесных пожаров → lentosammutus
авиационно-химические методы ухода и защиты → lentolevitys
автогамия → itsepölytys

Figure 7: Target-language index is displayed in a new window

This was the terminographic processor MyTerMS in a nutshell. Many improvements and enhancements are still planned. Among the most urgent ones are swapping the source and the target language (multi-directionality), locking of the entries being edited and keeping a history of searches. The glossary of the E.U. Neighbourhood Programme terms which has been compiled in the Common Language Project will be the first trilingual glossary to be stored in MyTerMS. There have also been plans to use MyTerMS as a starting point for a multilingual termbank of the Department of Translation Studies of the University of Helsinki. Time will show whether these plans become reality.

The users and uses of TshwaneLex One

Gilles-Maurice de Schryver & David Joffe

TshwaneDJe Human Language Technology, Pretoria, South Africa
Department of African Languages and Cultures, Ghent University, Belgium
E-mail: {gillesmaurice.deschryver,david.joffe}@tshwanedje.com
Web: http://tshwanedje.com/

Abstract Ten months after the release of the dictionary compilation software TshwaneLex 1.0, and just days away from the launch of TshwaneLex 2.0, this paper presents a snapshot of the various users and uses of TshwaneLex to date.

1. Introduction

Development of the commercial, off-the-shelf dictionary compilation software TshwaneLex began in May 2002. This followed an in-depth study of the then-available packages for and approaches to dictionary compilation, as well as a survey of lexicographers' dreams with regard to 'the dictionary of the future' (De Schryver 2003). During the development of TshwaneLex, early adopters included numerous teams in especially Africa and Europe. Following the launch of TshwaneLex 1.0 in September 2005, the client base quickly grew to well over a hundred users. Since then, seven free upgrades within the version one range have been released. Now, in June 2006, just days away from the launch of TshwaneLex 2.0, it seems like an appropriate moment to take stock of the users and uses of 'TshwaneLex One'.

Two cautionary notes are in order. Firstly, the current field report will to some extent be self-censored, as commercial clients typically do not wish to divulge their plans until their products have reached the market. Secondly, as was the case with the early adopters, around 40% of the current users have already migrated to the next version, 'TshwaneLex Two', so some aspects of the latter will also be touched upon.

2. TshwaneLex in a nutshell

TshwaneLex is a dictionary writing system to compile *any* type of dictionary with, for *any* language(s). It is not a CQS (corpus-query system), DTP (desktop publishing) software, nor is it a 'generic XML editor'. Rather, the goal was to create a more specialised tool specifically to optimise and assist with dictionary compilation, and to be as 'user-friendly' as possible in that regard. Observe that this was the initial focus, and that CQS and DTP features as well as increased XML support have been steadily added as the client base has grown and clients have requested this or that feature. With the 'Web as Corpus' (Kilgarriff & Grefenstette 2003) in mind, direct links between TshwaneLex and Google text and image searches have for example been implemented, export options have multiplied with more possibilities for both online (e.g. one article per HTML page) and hardcopy (e.g. first/last lemma on each page in running header) options, while it is now also possible to import XML documents.

In TshwaneLex, a strict separation is made between the actual dictionary contents (the *data*), the structure of each article (the *dictionary grammar* or *DTD* (document type definition)), and the way those contents, given a certain structure, (may) look (the *formatting* or *style*). Each of those levels is fully customisable, with the data level further subdividable into unique and repetitive (*metalanguage*) material.

Among the many dictionary-compilation-specific features built into TshwaneLex are fully automated cross-reference integrity tracking and updating (Joffe *et al*. 2003), dynamic metalanguage customisation (De Schryver & Joffe 2005b), multidimensional lexicographic

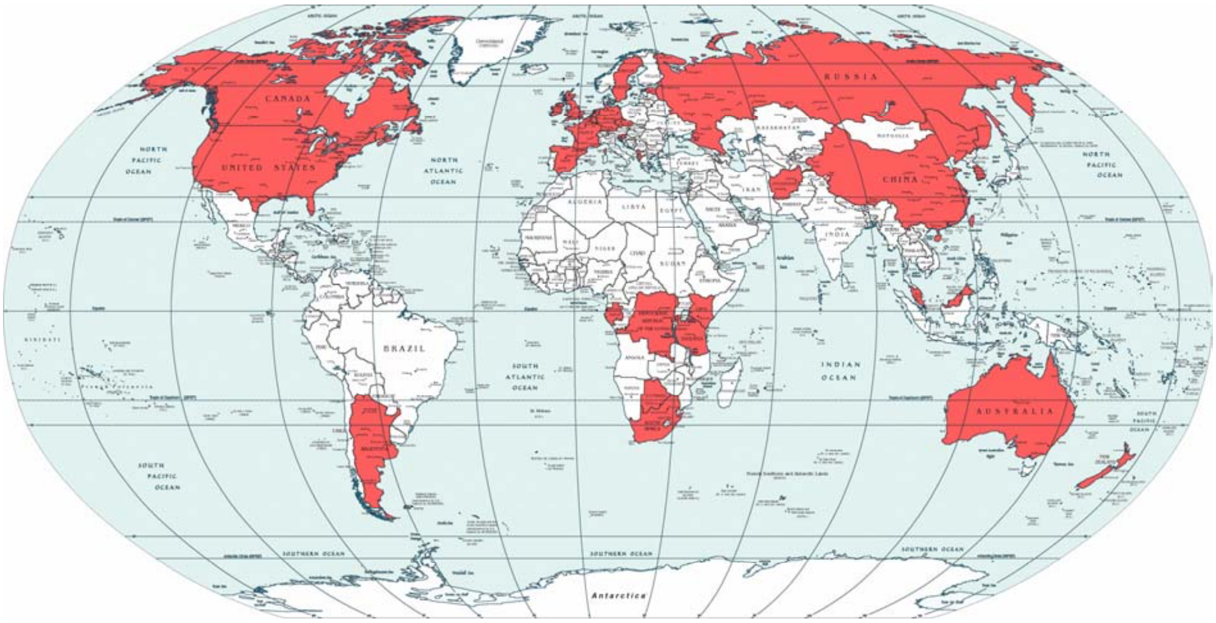
Rulers to help manage projects (De Schryver 2005), completely customisable sorting options (De Schryver & Joffe 2005a), etc., and specifically for bilingual and multilingual dictionary projects, powerful reversal and linked-view features (De Schryver & Joffe 2005a).

TshwaneLex One can be run from any stand-alone PC, and neither additional software nor knowledge of databases is required. (Note that porting to other platforms such as Mac and Linux is planned, while TshwaneLex Two contains network support.) Basically, TshwaneLex has its own 'internal format' for processing data in-memory which it always uses, and has a generic 'input/output layer' behind which backends/plugins exist (and more can be created) for loading/saving data from/to different underlying formats, including (1) the native TshwaneLex dictionary file (.tldict), (2) XML format, (3) a relational database, etc. (cf. the section 'extendible I/O architecture' in Joffe *et al*. (2003)). The 'internal format' can approximately be compared to a parsed XML document object. So internally TshwaneLex does not hold the data as XML, but rather, more like XML that has already been parsed into an in-memory structure. If saving to XML, the XML backend re-generates XML from the document object. If saving to a relational database, the relational database saves for instance rows to tables using SQL, and so on.

3. Basic user and usage statistics

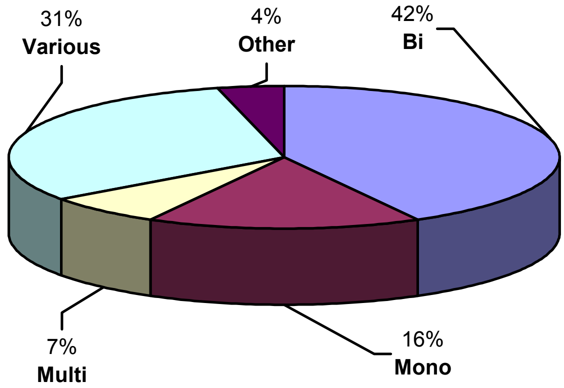
From the start, it has been the intention to cater for both commercial and academic projects, with in the latter case some level of philanthropy for languages listed in the *UNESCO Red Book of Endangered Languages*. With currently 195 users of TshwaneLex, the breakdown is as follows: 49% commercial, 47% academic, and 4% philanthropic. Around 29% of the users work on their own (in isolation), while 71% work on a project in group – in each case the software may be used to work on either a single or several projects simultaneously.

The family of TshwaneLex users presently spreads across the world, as is depicted on the following map:



TshwaneLex users are found in Afghanistan, Albania, Argentina, Australia, Belgium, Botswana, Canada, China, the Czech Republic, the Democratic Republic of the Congo, Estonia, France, Gabon, Germany, Ireland, Kenya, Luxembourg, Macao (China), Malaysia, the Netherlands, New Zealand, Russia, Rwanda, Slovenia, South Africa, South Korea, Spain, Sweden, Taiwan, Tanzania, the U.K., the U.S.A., and Wales (U.K.).

The number of different languages dealt with in TshwaneLex is even more diverse and currently approaches one hundred, among them: Afrikaans, Albanian, Alor Malay, Arabic, Acehnese, Bai, Balinese, Basque, Belarusian, Breton, Buginese, Bulgarian, Catalan, Chinese, Cilubà, Croatian, Czech, Danish, Dutch, East Javanese, English, Estonian, Finnish, French, Gaelic, German, Gimán, Haitian, Hmong, Iban, Icelandic, Indonesian, Inezeño Chumash, Irish, isiNdebele, isiXhosa, isiZulu, Italian, Jakarta Malay, Japanese, Javanese, Javindo, Kinyarwanda, Kiswahili, Korean, Kupang Malay, Ladino, Latin, Lingála, Low German, Macedonian, Madurese, Makassarian, Malay, Menadonesian, Minangkabau, Moluccan, Muna, Norwegian, Old English, Papiamento, Pashto, Petjoh, Picard, Polish, Polynesian, Portuguese, Romanian, Rotinese, Russian, Sahu, Sasak, Scots, Sesotho, Sesotho sa Leboa, Setswana, Singhalese, siSwati, Slovenian, Spanish, Sranantongo, Sundanese, Surinamese Javanese, Swedish, Terik, Ternate Malay, Tshivenda, Ukrainian, Virgin Islands Creole English, Walloon, Welsh, and Xitsonga.



Looking at the types of dictionaries that are being compiled with TshwaneLex, one notices that half the projects treat at least two languages (bilingual and semi-bilingual: 42%, trilingual and multilingual: 7%) versus only 16% that are truly monolingual. In every three out of ten projects (31%) a combination of types is being produced simultaneously, and in another 4% the focus is on the use of TshwaneLex to teach (meta)lexicography, to produce historical and dialect dictionaries or even pictionaries and encyclopaedias. Across the various types, roughly one fifth of the projects deal with LSP (language for specific purpose) dictionaries.

The extent/size of projects for which TshwaneLex is currently being used varies widely, from very small lexica to huge multi-volume reference works. To give an idea of a project between these extremes, the latest 1,552-page A4-size Afrikaans–English desktop dictionary by *Pharos* (Du Plessis *et al*. 2005) can easily be handled as a single TshwaneLex file on a single PC, with some statistics as follows:

- over 77,000 main entries;
- over 200,000 when including all sub-entries;
- over 2,400,000 elements (nodes) in the document tree;
- which corresponds to an 86MB TshwaneLex file;
- or exported as Unicode text, about 36MB;
- which translates to approximately 18 million characters.

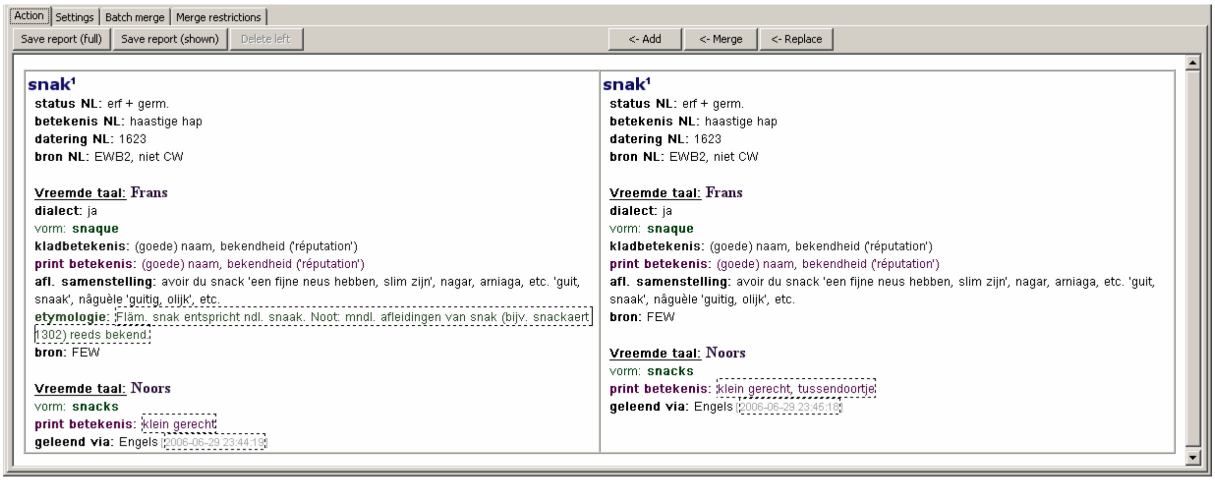
4. Sorts of issues arising in different sorts of circumstances

Clearly, with this wide geographical and typological coverage of users and uses, TshwaneLex simply had to support Unicode (as well as left-to-right and right-to-left scripts) on all levels. A flexible DTD with linked styles system that anyone *without programming* skills could set up also had to be, and *is*, part of the standard TshwaneLex package (Joffe & De Schryver 2005).

Perhaps surprisingly, the current needs did not include network support nor over-complex workflow modules. Conversely, all large teams (with on average around ten, but in one case up to thirty users) wished to have **advanced compare/merge tools** at their disposal. A special effort was therefore put into the development of these. When teams work in a distributed approach on a single project, three typical cases present themselves:

- different 'chunks' (e.g. different alphabetic sections, words belonging to different word classes, or even different semantic fields) are being worked on, and are simply merged periodically into a main database, after which that main database is redistributed to all compilers;
- each compiler focuses on certain aspects of each article only (phonetics, definitions, examples, etc.), with the same TshwaneLex file being sent from one compiler to the next. This approach is sometimes combined with the previous one;
- in especially multilingual setups, where up to a dozen languages are being worked on in parallel, each compiler focuses on his/her respective language(s), with their data then being merged periodically, and a new version of the main database being redistributed to all project members.

The latter approach is illustrated in the screenshot below, using data that is being prepared by a team of over a dozen compilers under the guidance of Nicoline van der Sijs, for her forthcoming book *Nederlands in de wereld*.



snak[1]
status NL: erf + germ.
betekenis NL: haastige hap
datering NL: 1623
bron NL: EWB2, niet CW

Vreemde taal: **Frans**
dialect: ja
vorm: **snaque**
kladbetekenis: (goede) naam, bekendheid ('réputation')
print betekenis: (goede) naam, bekendheid ('réputation')
afl. samenstelling: avoir du snack 'een fijne neus hebben, slim zijn', nagar, arniaga, etc. 'guit, snaak', nâguèle 'guitig, olijk', etc.
etymologie: Fläm. snak entspricht ndl. snaak. Noot: mndl. afleidingen van snak (bijv. snackaert 1302) reeds bekend.
bron: FEW

Vreemde taal: **Noors**
vorm: **snacks**
print betekenis: klein gerecht
geleend via: Engels 2006-06-29 23:44:19

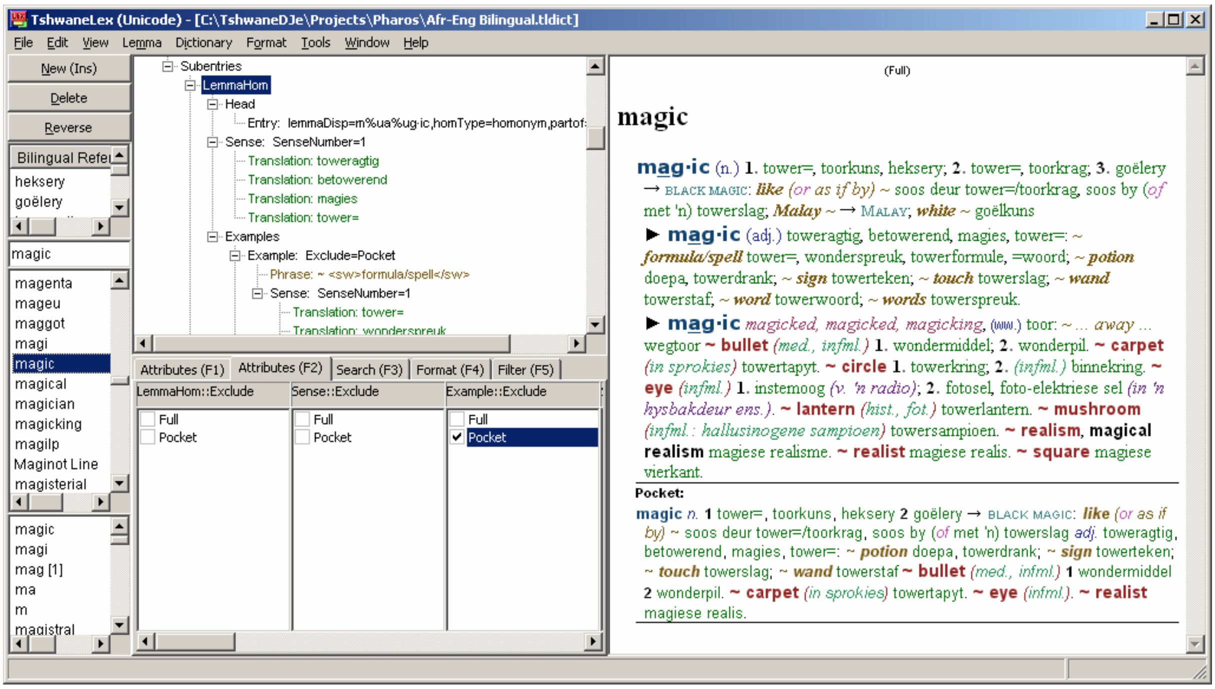
snak[1]
status NL: erf + germ.
betekenis NL: haastige hap
datering NL: 1623
bron NL: EWB2, niet CW

Vreemde taal: **Frans**
dialect: ja
vorm: **snaque**
kladbetekenis: (goede) naam, bekendheid ('réputation')
print betekenis: (goede) naam, bekendheid ('réputation')
afl. samenstelling: avoir du snack 'een fijne neus hebben, slim zijn', nagar, arniaga, etc. 'guit, snaak', nâguèle 'guitig, olijk', etc.
bron: FEW

Vreemde taal: **Noors**
vorm: **snacks**
print betekenis: klein gerecht, tussendoortje
geleend via: Engels 2006-06-29 23:45:18

In this project, an inventory is made of all the Dutch words that have entered other languages over the centuries. In TshwaneLex, each Dutch word has a treatment of its own (in the screenshot *snak* [homonym 1] 'snack' and the block immediately underneath it), and then linked to that any number of *Vreemde taal* 'foreign language' blocks (here, and so far, for French and Norwegian). In the compare/merge illustrated here, the data from the compiler focusing on the Nordic languages, which includes Norwegian, is being merged into the main database. In this case, the synonym *tussendoortje* will be added to *klein gerecht* which was already in the database. Observe that, in an earlier compare/merge pass, the updated/new material from the compiler focusing on French had already been added, and those changes will of course not be overwritten. Needless to say, after having clicked all the necessary merge restrictions, combining databases is a fully automatic and seamless process.

A second aspect that is becoming increasingly important is the notion to be able to 'extract' a multitude of dictionaries, each with their own characteristics, from a single, large database. Hence, with a single click, one typically wants to extract a pocket versus a desktop dictionary, or following another click a print edition versus an online version, or even a semi-bilingual versus a monolingual dictionary, all from the same TshwaneLex file, and in each case with the metalanguage in the appropriate format/language. In TshwaneLex this is achieved by means of allowing for **multiple sets of styles** to be set up, and in version two, a sophisticated new **'masks' feature**. These aspects are exemplified below for the above-mentioned *Pharos* dictionary, where several different dictionary projects are currently being integrated into one unique TshwaneLex file. (Note that in order not to divulge the publisher's plans, only two styles are shown here, 'Full' and 'Pocket'.)



Subentries
LemmaHom
Head
Entry: lemmaDisp=m%ua%ug·ic,homType=homonym,partof:
Sense: SenseNumber=1
Translation: toweragtig
Translation: betowerend
Translation: magies
Translation: tower=
Examples
Example: Exclude=Pocket
Phrase: ~ <sw>formula/spell</sw>
Sense: SenseNumber=1
Translation: tower=
Translation: wonderspreuk

LemmaHom::Exclude	Sense::Exclude	Example::Exclude
□ Full	□ Full	□ Full
□ Pocket	□ Pocket	☑ Pocket

(Full)

magic

mag·ic (n.) **1**. tower=, toorkuns, heksery; **2**. tower=, toorkrag; **3**. goëlery → BLACK MAGIC: ***like*** *(or as if by)* ~ soos deur tower=/toorkrag, soos by (*of* met 'n) towerslag; ***Malay*** ~ → MALAY; ***white*** ~ goëlkuns

▶ **mag·ic** (adj.) toweragtig, betowerend, magies, tower=: ~ ***formula/spell*** tower=, wonderspreuk, towerformule, =woord; ~ ***potion*** doepa, towerdrank; ~ ***sign*** towerteken; ~ ***touch*** towerslag; ~ ***wand*** towerstaf; ~ ***word*** towerwoord; ~ ***words*** towerspreuk.

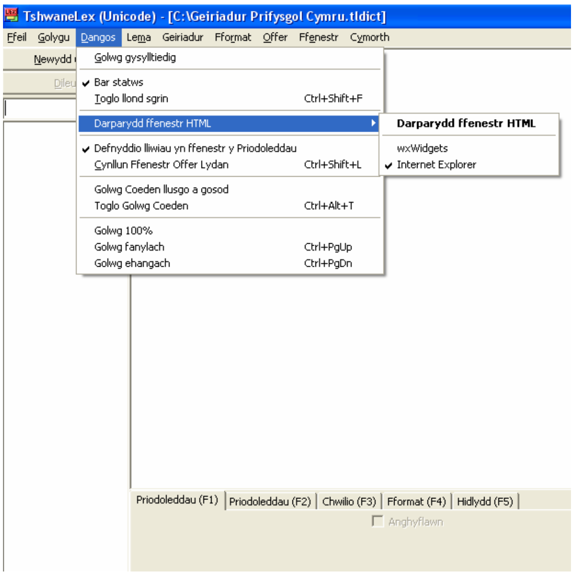
▶ **mag·ic** *magicked, magicked, magicking*, (ww.) toor: ~ ... *away* ... wegtoor **~ bullet** *(med., infml.)* **1**. wondermiddel; **2**. wonderpil. **~ carpet** *(in sprokies)* towertapyt. **~ circle** **1**. towerkring; **2**. *(infml.)* binnekring. **~ eye** *(infml.)* **1**. instemoog *(v. 'n radio)*; **2**. fotosel, foto-elektriese sel *(in 'n hysbakdeur ens.)*. **~ lantern** *(hist., fot.)* towerlantern. **~ mushroom** *(infml.: hallusinogene sampioen)* towersampioen. **~ realism**, **magical realism** magiese realisme. **~ realist** magiese realis. **~ square** magiese vierkant.

Pocket:

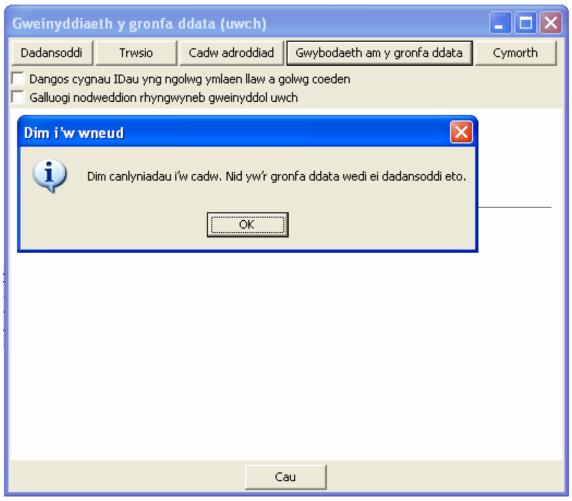
magic *n.* **1** tower=, toorkuns, heksery **2** goëlery → BLACK MAGIC: ***like*** *(or as if by)* ~ soos deur tower=/toorkrag, soos by (*of* met 'n) towerslag *adj.* toweragtig, betowerend, magies, tower=: ~ ***potion*** doepa, towerdrank; ~ ***sign*** towerteken; ~ ***touch*** towerslag; ~ ***wand*** towerstaf **~ bullet** *(med., infml.)* **1** wondermiddel **2** wonderpil. **~ carpet** *(in sprokies)* towertapyt. **~ eye** *(infml.)*. **~ realist** magiese realis.

In this screenshot, the option was chosen to display the various styles simultaneously in the preview area (the right half of the screen), so the various views (here 'Full' and 'Pocket') of every article can be seen concurrently. Note for example the different styles for lemma signs and parts of speech in the different editions, but also the automatic (re)numbering when outputting selected levels of the data.

Thirdly, and hardly surprising given that TshwaneLex is being used in all corners of the world, the wish was quickly voiced to enable the easy localisability of the GUI (graphical user interface). See in this regard the screenshots below for the (in-progress) translation into Welsh.



TshwaneLex (Unicode) - [C:\Geiriadur Prifysgol Cymru.tldict]
Ffeil Golygu Dangos Lema Geiriadur Fformat Offer Ffenestr Cymorth
Newydd
Dileu
Golwg gysylltiedig
✓ Bar statws
Toglo llond sgrin Ctrl+Shift+F
Darparydd ffenestr HTML
✓ Defnyddio lliwiau yn ffenestr y Priodoleddau
Cynllun Ffenestr Offer Lydan Ctrl+Shift+L
Golwg Coeden llusgo a gosod
Toglo Golwg Coeden Ctrl+Alt+T
Golwg 100%
Golwg fanylach Ctrl+PgUp
Golwg ehangach Ctrl+PgDn
Darparydd ffenestr HTML
wxWidgets
✓ Internet Explorer
Priodoleddau (F1)
Priodoleddau (F2)
Chwilio (F3)
Fformat (F4)
Hidlydd (F5)
□ Anghyflawn



Gweinyddiaeth y gronfa ddata (uwch)
Dadansoddi
Trwsio
Cadw adroddiad
Gwybodaeth am y gronfa ddata
Cymorth
Dim i'w wneud
OK
Cau

□ Dangos cygnau IDau yng ngolwg ymlaen llaw a golwg coeden
□ Galluogi nodweddion rhyngwyneb gweinyddol uwch

Dim canlyniadau i'w cadw. Nid yw'r gronfa ddata wedi ei dadansoddi eto.

© 2006 by Dewi Evans *et al*.

The localisation of the TshwaneLex GUI is put at every user's fingertips with the self-explanatory **built-in localisation editor**. Particularly handy is the fact that the results of the localisation can be seen in real time within TshwaneLex itself. At present, several localised versions of TshwaneLex are already in use in Asia, Europe and Africa, in among others (and respectively) Chinese, German and Cilubà.

5. Conclusion and outlook

Looking back, and keeping in mind that several dictionary writing systems did not quite make it in the past, the creation and distribution of TshwaneLex has become a true success story. Any licenses acquired henceforth will automatically be version two licenses. This second version of course includes everything the first version has, but also contains some significant improvements and a battery of new features. The already-mentioned ability to import XML, better network/multi-user support, and a more versatile approach to the concept of 'one database, many dictionaries', are some of them. Interlinked search features and filters, and numerous user interface improvements that help speed up compilation work, such as click-in-preview-to-edit, highlight selected element, or an optional pop-up window for work on long definitions, are but some of the others. No doubt, the family of TshwaneLex users will continue to grow, and with new users come new uses, and thus exciting new features.

References

De Schryver, G-M. 2003. Lexicographers' Dreams in the Electronic-Dictionary Age. *International Journal of Lexicography* 16.2: 143–199.

De Schryver, G-M. 2005. Concurrent Over- and Under-Treatment in Dictionaries – The *Woordeboek van die Afrikaanse Taal* as a Case in Point. *International Journal of Lexicography* 18.1: 47–75.

De Schryver, G-M & D. Joffe. 2005a. One database, many dictionaries – varying co(n)text with the dictionary application TshwaneLex. In Ooi, V.B.Y., A. Pakir, I. Talib, L. Tan, P.K.W. Tan & Y.Y. Tan (eds.). 2005. *Words in Asian Cultural Contexts, Proceedings of the 4th Asialex Conference, 1-3 June 2005, M Hotel, Singapore*: 54–59. Singapore: Department of English Language and Literature & Asia Research Institute, National University of Singapore.

De Schryver, G-M & D. Joffe. 2005b. Dynamic Metalanguage Customisation with the Dictionary Application TshwaneLex. In Kiefer, F., G. Kiss & J. Pajzs (eds.). 2005. *Papers in Computational Lexicography, COMPLEX 2005*: 190–199. Budapest: Linguistics Institute, Hungarian Academy of Sciences.

Du Plessis, M. *et al.* 2005. *Pharos Afrikaans–Engels / English–Afrikaans Woordeboek / Dictionary*. Cape Town: Pharos.

Joffe, D. & G-M de Schryver. 2005. Representing and describing words flexibly with the dictionary application TshwaneLex. In Ooi, V.B.Y., A. Pakir, I. Talib, L. Tan, P.K.W. Tan & Y.Y. Tan (eds.). 2005. *Words in Asian Cultural Contexts, Proceedings of the 4th Asialex Conference, 1-3 June 2005, M Hotel, Singapore*: 108–114. Singapore: Department of English Language and Literature & Asia Research Institute, National University of Singapore.

Joffe, D., G-M de Schryver & D.J. Prinsloo. 2003. Computational features of the dictionary application "TshwaneLex". *Southern African Linguistics and Applied Language Studies* 21.4 (Special issue on 'Human Language Technology in South Africa: Resources and Applications'): 239–250.

Kilgarriff, A. & G. Grefenstette. 2003. Special Issue on the Web as Corpus. *Computational Linguistics* 29.3: 333–502.

TshwaneLex. 2002–2006. Available from `http://tshwanedje.com/tshwanelex/`

CULLER – A user-friendly corpus query system

Elzbieta Dura

Lexware Labs, University of Skövde, Gothenburg, Sweden
E-mail: elzbieta@lexwarelabs.com
Web: http://www.nla.se/lexware/

Abstract Culler is a CQL (corpus query system) with focus on usability. Non-initiated users get a grip of the system within ten minutes. Even without mastering a language of regular expressions, one is able to formulate queries which yield precise lexico-grammatical extractions. The corpus tool is found useful not only by linguists but also by researchers in other disciplines. The simplicity of the interface does not necessarily mean that functionality is sacrificed. Culler provides, among other things, frequency-sorted extractions of freely chosen phrases and phrase patterns of any length, all in an instant. A function which is particularly useful in dictionary writing is the possibility to extract candidates for new dictionary words.

1. For corpus zealots and those who have not yet seen the light

The need to learn regular expressions discourages many potential users of corpora. And when they do try to learn they often get disheartened while using it. A missing semicolon can result in an insipid error message instead of a rich concordance.

Corpus tools are still primarily used by dedicated corpus-based linguists, which probably is the reason why user-friendliness has not been considered a major issue. Corpora seem to have a chance of becoming as widely used as dictionaries if the search process is simplified. Culler is a CQL (corpus query system) which focuses on usability and provides a user interface which is suitable for an amateur user.

A corpus specialist need not be disappointed, however. One of the particularly useful features present in Culler is the frequency-based sorting of text strings matching any search pattern. Another one is the possibility to extract candidates for new dictionary words or special language terms.

2. Basic facts about Culler

Culler has been developed by Lexware Labs in Gothenburg, Sweden. It is being used and tested by researchers and students at Göteborg University and Skövde University in Sweden and Kraków University in Poland. Culler is used in a browser, such as *Microsoft* Internet Explorer or *Mozilla* Firefox. It provides extractions of language data from very large corpora. The architecture of the system is typical of an information retrieval system, with the addition that text and request analyses are language specific, based on an explicit lexico-grammatical language representation. This fact is decisive for some query types (cf. section 6). English, Polish and Swedish instances have been created thus far.

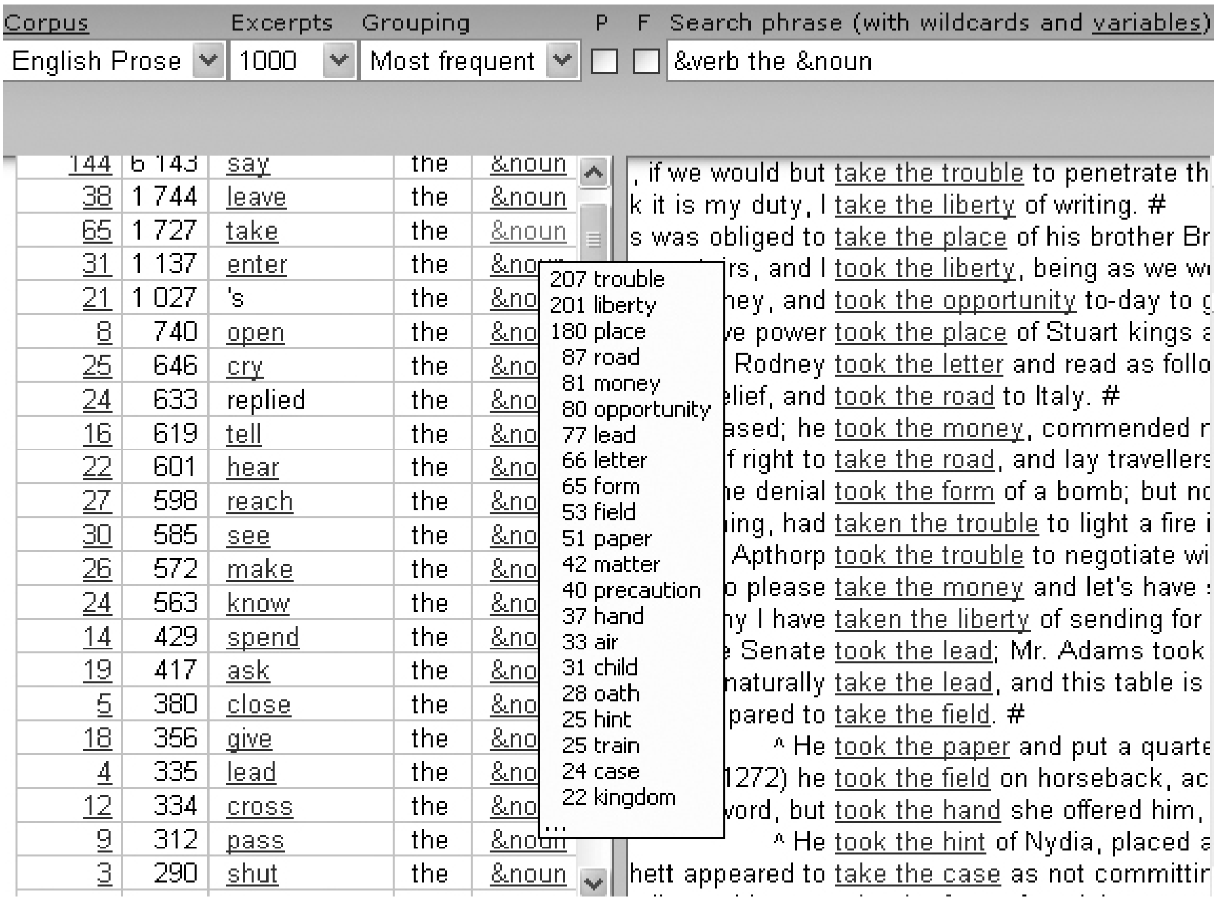
Culler is meant to work with several corpora and adding new corpora is easy. This feature is important, considering the need for ad hoc corpora in a variety of human activities. For instance, a corpus of texts including a mention of ‘stem cell’ was created for the Information Fusion Research Program (at University of Skövde) as a selection from PubMed (which is a database of citations from life sciences journals) – a Stem Cell Corpus. Another special corpus available in Culler is the corpus of Polish blogs (about 100 million words). The English, Polish and Swedish corpora present in the Culler demo at http://www.nla.se/

`culler/` consist of literary and journalistic texts which have been downloaded from the Internet. These three opportunistic corpora count about 50 million words each.

The dictionary of Culler's English instance is based on the WordNet dictionary. Simple, hand-coded word formation rules were used to identify inflected word forms. Part-of-speech disambiguation is based on the Stanford Tagger. The Polish Culler does not include part-of-speech disambiguation, only lemmatisation, based on the Polish Inflectional Dictionary. Swedish texts get the best analysis: in addition to lemmatisation and tagging, words are broken down into components. Analysis tools for Swedish are based on the Gothenburg Lexical Database.

3. User interface

The Culler window is split into three frames. The command frame is at the top, where a corpus query is entered and some options are set. Once the query has been executed, the matching strings are shown in the concordance frame on the right and in the table summarising the results on the left. Figure 1 shows the results for a query "&verb the &noun", which means a verb followed by the, followed by a noun ("&" marks a class of words).



Corpus	Excerpts	Grouping	P	F	Search phrase (with wildcards and variables)
English Prose	1000	Most frequent	□	□	&verb the &noun

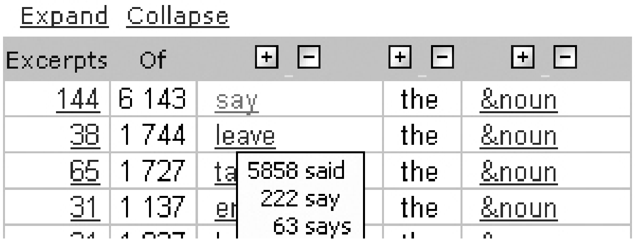
144	6 143	say	the	&noun
38	1 744	leave	the	&noun
65	1 727	take	the	&noun
31	1 137	enter	the	&no
21	1 027	's	the	&no
8	740	open	the	&no
25	646	cry	the	&no
24	633	replied	the	&no
16	619	tell	the	&no
22	601	hear	the	&no
27	598	reach	the	&no
30	585	see	the	&no
26	572	make	the	&no
24	563	know	the	&no
14	429	spend	the	&no
19	417	ask	the	&no
5	380	close	the	&no
18	356	give	the	&no
4	335	lead	the	&no
12	334	cross	the	&no
9	312	pass	the	&noun
3	290	shut	the	&noun

207	trouble
201	liberty
180	place
87	road
81	money
80	opportunity
77	lead
66	letter
65	form
53	field
51	paper
42	matter
40	precaution
37	hand
33	air
31	child
28	oath
25	hint
25	train
24	case
22	kingdom
...	

, if we would but take the trouble to penetrate th
k it is my duty, I take the liberty of writing. #
s was obliged to take the place of his brother Br
rs, and I took the liberty, being as we w
ney, and took the opportunity to-day to g
e power took the place of Stuart kings a
Rodney took the letter and read as follo
lief, and took the road to Italy. #
ased; he took the money, commended r
f right to take the road, and lay travellers
e denial took the form of a bomb; but nc
ing, had taken the trouble to light a fire i
Apthorp took the trouble to negotiate wi
o please take the money and let's have
y I have taken the liberty of sending for
Senate took the lead; Mr. Adams took
naturally take the lead, and this table is
pared to take the field. #
^ He took the paper and put a quarte
1272) he took the field on horseback, ac
vord, but took the hand she offered him,
^ He took the hint of Nydia, placed a
hett appeared to take the case as not committir

Figure 1: Results for "&verb the &noun" with a tooltip for "&noun" at "take"

"Grouping" is an important parameter in the system. When it is set to "Most frequent" the retrieved distinct phrases are sorted according to their frequency of occurrence in the whole corpus. The summary table in Figure 2 shows frequency sorted phrases of the type: a verb followed by "the" followed by a noun. The most frequent verb in this context is "say" in the English corpus of the demo. The first column of the summary table shows the number of distinct excerpts retrieved into the concordance, the second column shows the number of matching excerpts in the whole corpus (this number is not always equal to the total number of occurrences, cf. section 5).



Expand Collapse

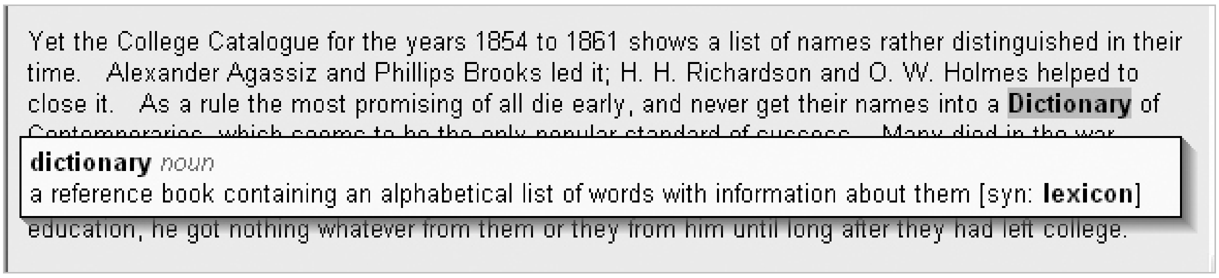
Excerpts	Of	⊞ ⊟	⊞ ⊟	⊞ ⊟
144	6 143	say	the	&noun
38	1 744	leave	the	&noun
65	1 727	ta	the	&noun
31	1 137	er	the	&noun

5858 said
222 say
63 says

Figure 2: A tooltip prompting about all forms of “say” present in extractions

Three detail levels can be viewed in the summary table: word class, lemma and word form. A quick way to browse the details is to draw a cursor over a variable or a lemma, upon which a tooltip appears. The lowest detail level for the whole table is shown when “Expand” is clicked. Clicking on a plus sign reveals the details of a column; to get back to the broad view “Collapse” or a minus sign can be used. Changing the order in which the three levels are exposed provides a variety of co-occurrence statistics, e.g. “&verb the trouble”, or “say the &noun”, etc.

A broader context can be obtained by clicking the underlined part of a concordance line. The context window is unusual in that dictionary definitions are hidden behind the words. They appear in a tooltip on clicking a word, as shown in Figure 3.



Yet the College Catalogue for the years 1854 to 1861 shows a list of names rather distinguished in their time. Alexander Agassiz and Phillips Brooks led it; H. H. Richardson and O. W. Holmes helped to close it. As a rule the most promising of all die early, and never get their names into a **Dictionary** of Contemporaries, which seems to be the only popular standard of success. Many died in the war.

dictionary *noun*
a reference book containing an alphabetical list of words with information about them [syn: **lexicon**]

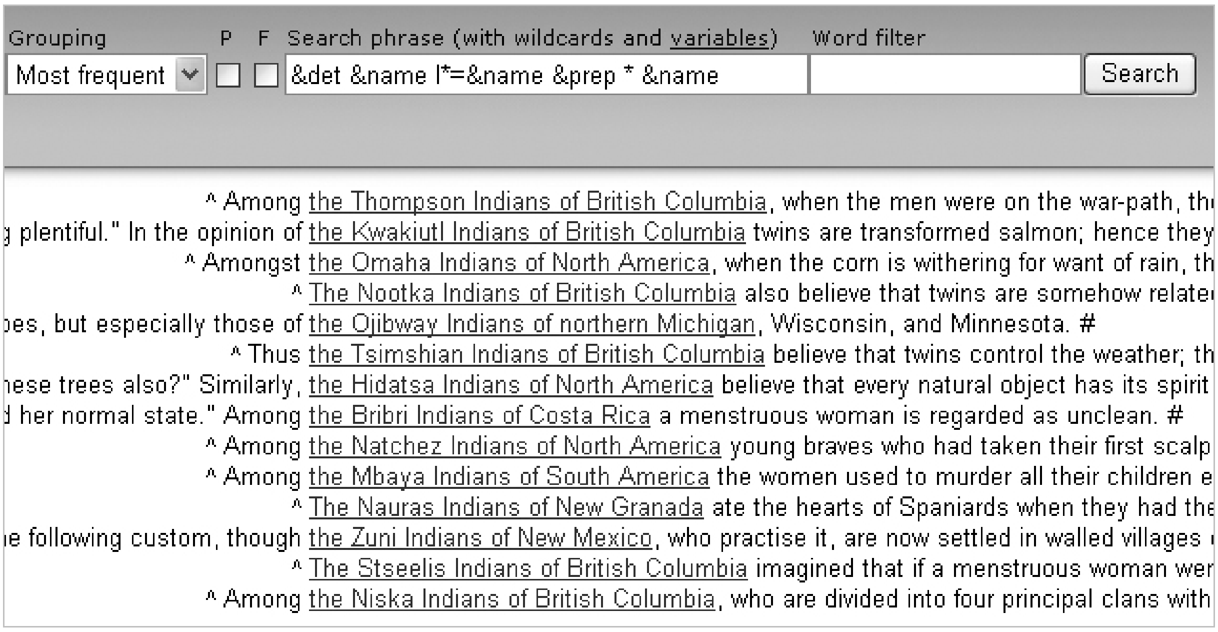
education, he got nothing whatever from them or they from him until long after they had left college.

Figure 3: Dictionary definitions appear in the window with broader context

4. Queries

Query formulation is easily learnt by trying the examples in the Culler tutorial. The user is able to master the query language gradually, from simple to complex queries. A search phrase is interpreted as a sequence of words, or symbols for word classes. Wildcards can also be used, and symbols may be selected from a list. A word in a search phrase is matched in all its forms unless it is written in quotes. Figure 4 shows an example of possible combinations of symbols: “I*=&name” means ‘any name beginning with an I’. Despite the simplicity of the query language, precise queries can be formulated. Combined with frequency-based sorting, this leads to a variety of useful selections, such as set phrases and collocations.

Keeping the query language simple may involve some limitations in querying. For instance, the intuitive interpretation of a search phrase as a sequence of words does not allow the expression of intervals of positions. For example, when looking for an adjective in positions 1-3 before a noun, three separate queries need to be entered: “&adj &noun”, “&adj * &noun”, “&adj * * &noun”.



Grouping: Most frequent ☐ ☐

Search phrase (with wildcards and variables): &det &name I*=&name &prep * &name

Word filter:

```
                              ^ Among the Thompson Indians of British Columbia, when the men were on the war-path, th
g plentiful." In the opinion of the Kwakiutl Indians of British Columbia twins are transformed salmon; hence they
                           ^ Amongst the Omaha Indians of North America, when the corn is withering for want of rain, th
                                   ^ The Nootka Indians of British Columbia also believe that twins are somehow relate
oes, but especially those of the Ojibway Indians of northern Michigan, Wisconsin, and Minnesota. #
                                ^ Thus the Tsimshian Indians of British Columbia believe that twins control the weather; th
hese trees also?" Similarly, the Hidatsa Indians of North America believe that every natural object has its spirit
d her normal state." Among the Bribri Indians of Costa Rica a menstruous woman is regarded as unclean. #
                              ^ Among the Natchez Indians of North America young braves who had taken their first scalp
                              ^ Among the Mbaya Indians of South America the women used to murder all their children e
                                   ^ The Nauras Indians of New Granada ate the hearts of Spaniards when they had the
e following custom, though the Zuni Indians of New Mexico, who practise it, are now settled in walled villages
                                   ^ The Stseelis Indians of British Columbia imagined that if a menstruous woman wer
                              ^ Among the Niska Indians of British Columbia, who are divided into four principal clans with
```

Figure 4: A concordance obtained for the query "&det &name I*=&name &prep * &name"

5. Frequency-based sorting

The most valuable feature is the possibility of obtaining frequency-ordered selections of freely defined word sequences in an instant. Even very general queries such as the most frequent *n*-word sequences in the whole corpus are answered within seconds.

This feature was diligently exploited in extracting constructions which are specific to the language of biomedical articles from the Stem Cell Corpus (Dura *et al*. 2006). Frequency sorting helps bring multiword terminology to the surface, as shown in Table 1 – a part of the summary table for the query "* &noun &noun &noun &noun #" (the last symbol means the end of a sentence).

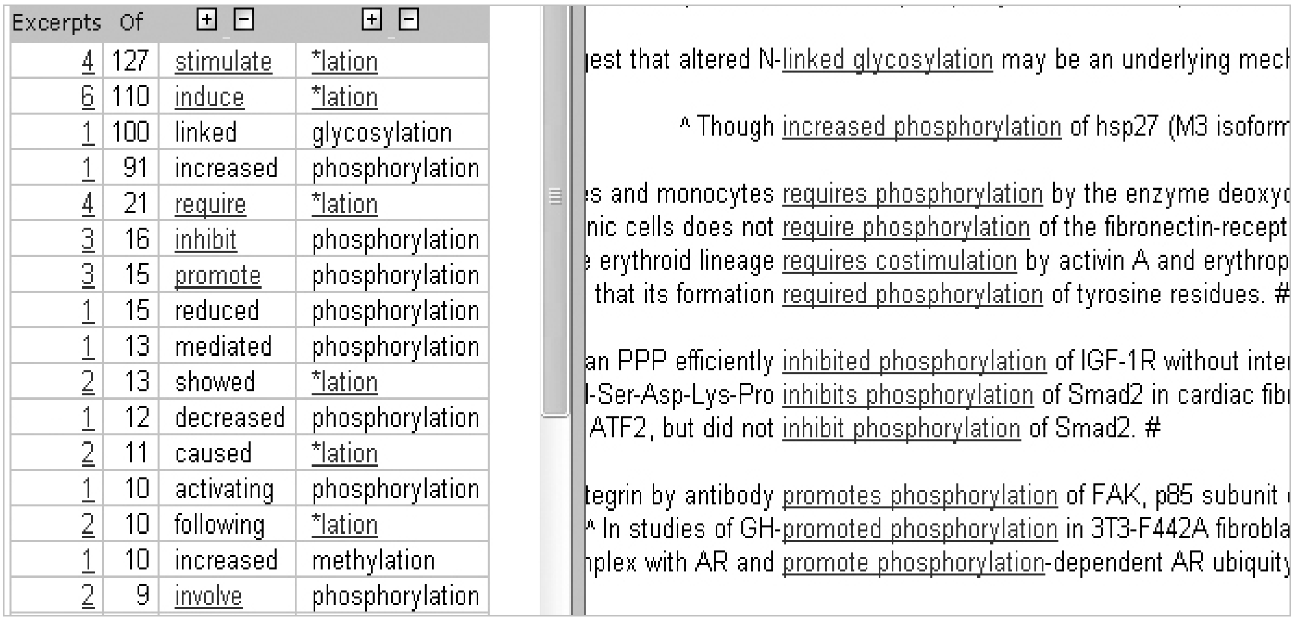
Table 1: Frequency-sorted extraction of sequences of four nouns from the Stem Cell Corpus

1	406	peripheral	blood	stem	cell	transplantation	#
1	107	reverse	transcriptase	polymerase	chain	reaction	#
1	95	reverse	transcription	polymerase	chain	reaction	#
1	91	long	term	bone	marrow	cultures	#
1	58	hematopoietic	stem	cell	transplant	recipients	#
1	57	peripheral	blood	progenitor	cell	transplantation	#

The second column of a summary table discloses the number of matching excerpts which need to be consulted in order to provide a frequency-sorted selection. These numbers are not always equal to the total number of occurrences of an excerpt in the corpus; this occurs only when the number of distinct excerpts desired by a user (parameter "Excerpts") is higher than the number of occurrences shown in the second column. However, the frequency ordering is always valid irrespective of how the "Excerpts" parameter is set, that is, top frequent matches come always first.

6. Extracting candidates for lexical entries

Indexing in Culler is done with reference to a specific dictionary. This feature enables queries on candidates for dictionary words. In order to provide a valuable selection of this kind, Culler excludes not only words already present in the dictionary but also non-dictionary words such as proper names. Combined with frequency ordering, selections of this type can be useful in updating general dictionaries or in extracting terms of a special language. Figure 5 shows a selection of verbs co-occurring most often with deverbal nouns from the Stem Cell Corpus. The query used here is "&verb *lation=&new", meaning any verb followed by a word which is not present in the general dictionary and ends in -lation.



Excerpts	Of	⊞ ⊟	⊞ ⊟
4	127	stimulate	*lation
6	110	induce	*lation
1	100	linked	glycosylation
1	91	increased	phosphorylation
4	21	require	*lation
3	16	inhibit	phosphorylation
3	15	promote	phosphorylation
1	15	reduced	phosphorylation
1	13	mediated	phosphorylation
2	13	showed	*lation
1	12	decreased	phosphorylation
2	11	caused	*lation
1	10	activating	phosphorylation
2	10	following	*lation
1	10	increased	methylation
2	9	involve	phosphorylation

```
jest that altered N-linked glycosylation may be an underlying mecl

          ^ Though increased phosphorylation of hsp27 (M3 isoforr

s and monocytes requires phosphorylation by the enzyme deoxyc
nic cells does not require phosphorylation of the fibronectin-recept
erythroid lineage requires costimulation by activin A and erythrop
 that its formation required phosphorylation of tyrosine residues. #

an PPP efficiently inhibited phosphorylation of IGF-1R without inte
l-Ser-Asp-Lys-Pro inhibits phosphorylation of Smad2 in cardiac fib
 ATF2, but did not inhibit phosphorylation of Smad2. #

tegrin by antibody promotes phosphorylation of FAK, p85 subunit
^ In studies of GH-promoted phosphorylation in 3T3-F442A fibrobla
nplex with AR and promote phosphorylation-dependent AR ubiquity
```

Figure 5: Frequency-ordered sequences of a verb followed by a "new" deverbal noun

7. Friendly and functional

The smooth entry to the system resides partly in the fact that default values are set for all parameters and that a search phrase is simply interpreted as a sequence of words. When query formulation is reduced to entering a sequence of words or symbols for classes of words, even the first-time user is bound to succeed in obtaining some concordance.

The usefulness of Culler in dictionary writing is the result of two features. One is the fact that queries can be related to a specific dictionary, which is represented explicitly in the system. The other one is that frequency-sorting is possible for all kinds of queries. These features provide for selections such as: the most frequent "new" words or the most frequent terms of a special language.

References

A. Dictionaries

Gothenburg Lexical Database. http://spraakbanken.gu.se/
Polish Inflectional Dictionary. http://winnie.ics.agh.edu.pl/proj_uk/fleksbaz/
WordNet. http://wordnet.princeton.edu/

B. Other literature

Dura, E., B. Erlendsson, B. Gawronska & B. Olsson. *forthcoming* in 2006. Towards Information Fusion in Pathway Evaluation: Encoding Relations in Biomedical Texts. In *Proceedings of The 9th International Conference on Information Fusion*. Florence: ISIF.

Information Fusion Research Program. `http://www.infofusion.se/`

PubMed. `http://www.ncbi.nlm.nih.gov/entrez/query/static/overview.html`

Stanford Tagger. `http://nlp.stanford.edu/software/tagger.shtml`

Particulars of Presenters and Participants

Andrea **Abel**
✉ Institute for Specialised Communication and Multilingualism, European Academy
Drususallee/Viale Druso, 1
Bozen/Bolzano
39100
Italy
⌨ andrea.abel@eurac.edu
💻 http://www.eurac.edu/Org/LanguageLaw/Multilingualism/index.htm

★

Xabier **Artola-Zubillaga**
✉ Faculty of Computer Science, University of the Basque Country
Manuel de Lardizabal Pasealekua, 1
Donostia-San Sebastián (Gipuzkoa)
20018
Spain
⌨ xabier.artola@ehu.es
💻 http://www.sc.ehu.es/siwebso/Principal/principal-i.htm

★

Mikel **Astiz**
✉ Faculty of Computer Science, University of the Basque Country
Manuel de Lardizabal Pasealekua, 1
Donostia-San Sebastián (Gipuzkoa)
20018
Spain
⌨ mastiz001@ikasle.ehu.es
💻 http://www.sc.ehu.es/siwebso/Principal/principal-i.htm

★

David **Bodine**
✉ Apartado 39
Algeciras
11200
Spain
⌨ asigwan@yahoo.com

★

Willem **Botha**
⌨ Bureau of the Woordeboek van die Afrikaanse Taal (WAT)
P.O. Box 245
Stellenbosch
7599
South Africa
⌨ wfb@sun.ac.za
💻 http://www.sun.ac.za/wat/

★

Stefano **Bracco**
✉ Information & Communication Technologies, European Academy
Drususallee/Viale Druso, 1
Bozen/Bolzano
39100
Italy
⌨ stefano.bracco@eurac.edu
💻 http://www.eurac.edu/About/ServiceDep/ict.htm

★

Dino **Bressan**
✉ Department of French and Italian Studies, University of Melbourne
Parkville, Victoria
Melbourne
3010
⌨ dino@unimelb.edu.au
💻 http://www.french-italian.unimelb.edu.au/

★

Carmela **Chateau**
✉ Centre des Sciences de la Terre, University of Burgundy
6 Bd Gabriel
Dijon
21000

France
⌨ carmela.chateau@u-bourgogne.fr
💻 http://www.u-bourgogne.fr/

★

Isabella **Chiari**
✉ Università “La Sapienza” di Roma
Via dei Genovesi 36
Rome
00153
Italy
⌨ isabella.chiari@uniroma1.it
💻 http://www.alphabit.net/

★

Ji-myoung **Choi**
✉ Neungyule Education, Inc.
Poongsung Bldg. 9F., 447-5 Seokyo-dong, Mapo-gu
Seoul
121-841
South Korea
⌨ amancio@neungyule.com
💻 http://www.neungyule.com/

★

Philippe **Climent**
✉ Ingénierie Diffusion Multimédia (IDM)
27 Rue Albert Einstein
B.P. 117 - Champs sur Marne
Marne la Vallée Cedex 2
77423
France
⌨ climent@idm.fr
💻 http://www.idm.fr/

★

Elena **Dal Pra**
✉ Arnoldo Mondadori Editore
Via Mondadori 15
Verona
37131
Italy
⌨ reference@mondadori.it
💻 http://www.mondadori.com/index_en.jsp

★

Gilles-Maurice **de Schryver**
✉ TshwaneDJe Human Language Technology
P.O. Box 299
Wapadrand, Pretoria
0050
South Africa
⌨ gillesmaurice.deschryver@tshwanedje.com
💻 http://tshwanedje.com/

★

Elzbieta **Dura**
✉ Lexware Labs, University of Skövde
Bagskyttebacken 4
Gothenburg
41319
Sweden
⌨ elzbieta@lexwarelabs.com
💻 http://www.nla.se/lexware/

★

Cristiano **Furiassi**
✉ Università degli Studi di Torino
Via Sant'Ottavio 20
Turin
10124
Italy
⌨ cristiano.furiassi@unito.it
💻 http://www.unito.it/

★

Antton **Gurrutxaga**
✉ Elhuyar Foundation
Zelai Haundi, 3 Osinalde Industrialdea
Usurbil
20170
Spain
⌨ agurrutxaga@elhuyar.com

★

Kristina **Hmeljak**
✉ Department for Asian and African Studies, University of Ljubljana

Askerceva 2
Ljubljana
1000
Slovenia
⌨ kristina.hmeljak@guest.arnes.si
🖳 http://www.uni-aas.si/index.php?jezik=en

★

Aleš **Horák**
🖃 Faculty of Informatics, Masaryk University
Botanicka 68a
Brno
602 00
Czech Republic
⌨ hales@fi.muni.cz
🖳 http://www.fi.muni.cz/

★

David **Joffe**
🖃 TshwaneDJe Human Language Technology
P.O. Box 299
Wapadrand, Pretoria
0050
South Africa
⌨ david.joffe@tshwanedje.com
🖳 http://tshwanedje.com/

★

Madis **Jürviste**
🖃 Association franco-estonienne de lexicographie
Pärna 4-34
Tartu
50603
Estonia
⌨ madis@jyrviste.fr
🖳 http://www.estfra.ee/Home.po

★

Varvara **Karzi**
🖃 Valtetsiou 14
Athens
11472
Greece
⌨ vkarzi@patakis.gr

★

Adam **Kilgarriff**
🖃 Lexical Computing Ltd.
71 Freshfield Rd.
Brighton
BN2 0BL
United Kingdom
⌨ adam@lexmasterclass.com
🖳 http://www.kilgarriff.co.uk/

★

Wilma **Kluiver**
🖃 Van Dale Lexicografie bv
St. Jacobsstraat 127
Utrecht
3511 BP
Netherlands
⌨ wilma.kluiver@vandale.nl
🖳 http://www.vandale.nl/

★

Igor **Kudashev**
🖃 Palmenia Centre for Continuing Education, University of Helsinki
PL 239, Paraatikentta 7
Kouvola
45101
Finland
⌨ igor.kudashev@helsinki.fi
🖳 http://www.helsinki.fi/palmenia/english/

★

Irina **Kudasheva**
🖃 Palmenia Centre for Continuing Education, University of Helsinki
PL 239, Paraatikentta 7
Kouvola
45101
Finland
⌨ irina.kudasheva@helsinki.fi
🖳 http://www.helsinki.fi/palmenia/english/

★

Margit **Langemets**
✉ Institute of the Estonian Language
Roosikrantsi 6
Tallinn
10119
Estonia
⌨ margit.langemets@eki.ee
💻 http://www.eki.ee/index.html.en

★

Andres **Loopmann**
✉ Institute of the Estonian Language
Roosikrantsi 6
Tallinn
10119
Estonia
⌨ andres.loopmann@eki.ee
💻 http://www.eki.ee/index.html.en

★

Michal **Mechura**
✉ Fiontar, Dublin City University
Collins Avenue
Dublin
9
Ireland
⌨ valselob@hotmail.com
💻 http://www.dcu.ie/fiontar/index.shtml

★

Jane **Nystedt**
✉ Institutionen för franska, italienska och klassiska språk, Stockholm University
Stockholm
10691
Sweden
⌨ jane.nystedt@fraita.su.se
💻 http://www.fraita.su.se/

★

Karel **Pala**
✉ Faculty of Informatics, Masaryk University
Botanicka 68a
Brno
602 00
Czech Republic
⌨ pala@fi.muni.cz
💻 http://www.fi.muni.cz/

★

Jan **Pomikalek**
✉ Faculty of Informatics, Masaryk University
Botanicka 68a
Brno
602 00
Czech Republic
⌨ xpomikal@fi.muni.cz
💻 http://www.fi.muni.cz/

★

Adam **Rambousek**
✉ Faculty of Informatics, Masaryk University
Botanicka 68a
Brno
602 00
Czech Republic
⌨ xrambous@fi.muni.cz
💻 http://www.fi.muni.cz/

★

Pavel **Rychlý**
✉ Faculty of Informatics, Masaryk University
Botanicka 68a
Brno
602 00
Czech Republic
⌨ pary@fi.muni.cz
💻 http://www.fi.muni.cz/

★

Xabier **Saralegi**
✉ Elhuyar Foundation
Zelai Haundi, 3 Osinalde Industrialdea
Usurbil
20170
Spain
⌨ xabiers@elhuyar.com

★

Peter **Steffensen**
✉ Knowledge Communication Lab,
Handelshøjskolen i Århus
Fuglesangsallé 4
Århus V
8210
Denmark
⌨ pst@asb.dk
💻 http://www.asb.dk/research/centresteams/centres/kcl.aspx

★

Marja **Verburg**
✉ Van Dale Lexicografie bv
St. Jacobsstraat 127
Utrecht
3511 BP
Netherlands
💻 http://www.vandale.nl/

★

Silvia **Verdiani**
✉ Zanichelli
Via Cernaia 1
Turin
10121
Italy
⌨ s.verd@inwind.it
💻 http://www.zanichelli.it/

★

Ülle **Viks**
✉ Institute of the Estonian Language
Roosikrantsi 6
Tallinn
10119
Estonia
⌨ ylle.viks@eki.ee
💻 http://www.eki.ee/index.html.en

★

Eveline **Wandl-Vogt**
✉ Institute of Lexicography of Austrian
Dialects and Names, Austrian Academy
of Sciences
Postgasse 7/1/2
Vienna
1010
Austria
⌨ eveline.wandl-vogt@oeaw.ac.at
💻 http://www.oeaw.ac.at/dinamlex/