DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool

Aleš Horák and **Karel Pala** and **Adam Rambousek** and **Martin Povolný**
Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic
{hales,pala,xrambous,xpovolny}@fi.muni.cz

Abstract

In this paper, we present the new wordnet development tool called DEBVisDic. It is built on the recently developed platform for client-server XML databases, called DEB II. This platform is able to cover many possible applications, from which we concentrate on the new, complete reimplementation of one of the most-spread wordnet editor and browser – VisDic.

We argue for the benefits the new DEB II platform brings to wordnet editing and to XML databases in general. In the paper, we describe the state of the implementation, the insides and interfaces of the DEBVisDic tool. We also discuss its functionality and some distinctions in comparison with other dictionary writing systems.

1 Introduction

The original wordnet, Princeton WordNet, is one of the most popular lexical resources in the NLP field [Fellbaum, 1998]. It was followed by multilingual EuroWordNet 1, 2 projects (1998-99) [Eur,] and Balkanet project (2001-4) [Bal,] 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 this work the software tools for browsing and editing wordnets have been designed and implemented, without whose the job could hardly have been performed. The first browser developed at Princeton is still used there and can be found at `http://www.cogsci.princeton.edu/~wn/` or `http://charity.princeton.edu`, within the EuroWordNet project the Polaris (and Periscope) tools have been implemented and used [M., 1998].

For Balkanet project the browser and editor VisDic (works both on Windows and Linux platform and can be downloaded from the page `http://nlp.fi.muni.cz/projects/visdic/`) has been prepared at the NLP Laboratory at the Faculty of Informatics Masaryk University [Horák and Smrž, 2003] since the development of the Polaris tool has been closed by 1999.

In comparison with the previous tools VisDic exploits XML data format thus making the wordnet-like databases more standard and exchangeable. Not only that, thanks to the XML data format used and to its dictionary specific configurability VisDic can serve for developing various types of dictionaries, i.e. monolingual, translational, thesauri and multilingually linked wordnet-like databases. The experience with the VisDic tool during Balkanet project has been positive [Horák and Smrž, 2004] and it was used as the main tool with which all Balkanet wordnets were developed.

VisDic, however, has its disadvantages, particularly it is not based on the client/server architecture and it does not allow to associate various attributes with literals and handle the links between them. It can work with links only between synsets which is a limiting feature for enriching wordnets with various sorts of information, e.g. in Czech with word derivation relations existing within one part of speech as well as across them.

The experience with VisDic has led us to more systematic research into the usage of XML data formats within the field of the computational lexicography. In parallel, we also pay attention to the relations between wordnets and Semantic Web. This interest gives us a strong motivation for studying the properties of the XML data formats and tools for working with them.

Thus we set as our task to design and implement a more universal dictionary writing system that could be exploited in various lexicographic applications to build large lexical databases. The system has been called Dictionary Editor and Browser (further DEB) [Smrž and Povolný, 2003] and its final version named DEB II will be used as a main tool for development of the Czech Lexical Database in close cooperation with the Institute of Czech Language, Czech Academy of Sciences, Prague. The design of the DEB allows us to modify it also for building wordnet-like databases. In this paper we are pleased to present the first results of this effort, i.e. the new tool for browsing and editing wordnet-like databases called DEBVisDic, a successor of VisDic.

During EuroWordNet projects another tool for WordNet editing has been developed in Universitat Politècnica de Catalunya, the tool named Web EuroWordNet Interface, WEI [Benítez et al., 1998]. This tool allows to edit wordnets with the CGI-script interface, which means that all the data is processed only on the server and users work with thin clients capable of HTML web browsing. All the data is stored in miniSQL database which is used as a direct storage place for the original XML data. Setting aside the fact that the WEI project seems finished further development in 2000, in comparison with the WEI tool, the presented DEB II

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 325–328.

platform has several advantages. Thanks to the underlying Mozilla engine, the DEB II platform is very strictly based on current W3C standards (XML, RDF, XUL, ...) and its design keeps strict separation of application layout and code. The client-server communication in DEB is based on firmly set standards (HTTP and SOAP). Last but not least, the DEB clients are not pure thin clients, thus making it possible to speed up the user interaction in offline processing.

2 The DEB Development Platform

The DEB platform (DEB II, 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 allows a modular composition of all services.

The clients communicate with servlets using HTTP requests in a manner similar to recently popular concept in web development called AJAX (Asynchronous JavaScript and XML [Rosenfeld and Morville, 1998]) or using the SOAP protocol [soa,]. The data are transported (using plain HTTP) in RDF, generic XML or plain-text formats or are marshalled using SOAP.

The actual data storage backend on the server side is provided by 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 widely-used Berkeley DB embedded database which runs on many system and devices ranging from Linux and Windows operating systems to mobile phones. Berkeley DB XML comes in form of a C++ library with interfaces to many scripting languages.

Since the client applications are mostly oriented to the graphical user interfaces (GUI), we have decided to adopt the concepts of the Mozilla Development Platform [Oeschger, 2002]. The Mozilla platform provides a complete set of tools for software development. Firefox web browser is one of the many applications created using this platform. Other applications include Mozilla Thunderbird mail client, Netscape web browser, Komodo integrated development environment or Nvu web page editor.

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 the possibility of concurrent work of different team members which leads to saving time.

Mozilla platform is open source free software which ensures that it will stay free and its development will continue. Every new major version adds more features and possibilities. Also, thanks to open source design, there's large number of free extensions of existing applications or the platform itself. Mozilla developers pay very much attention to security and the bugs are usually fixed very quickly.

Applications built on the Mozilla platform are working within many operating systems, actually any OS on which Mozilla runs (i.e. officially Windows, Linux, and Mac OS X, unofficially many others).

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 relatively simple creation of cross platform applications with possibility of easy customization of design, texts and localization. XUL itself is aimed only on creation of user interface, e.g. windows, buttons or toolbars, but it incorporates wide range of standardized technologies.

2.1 The DEB Server Side

The server side of DEB is implemented in the programming language called Ruby. Ruby (originating in Japan) is an object-oriented, interpreted programming language with week type checking. The DEB server uses also various additional libraries, both pure Ruby and interfaces to C/C++ libraries.

The DEB server suite runs on Linux, currently it is tested with Ubuntu Hoary on x86 and AMD64 architectures, but it should generally run on any recent UNIX-based system (including Mac OS X).

2.2 Usage Variability – Current DEB Clients

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

The particular DEB clients that are currently being implemented within the DEB platform include:

- DEBVisDic – new version of wordnet editor and browser, see the Section 3,
- DEBDICT – general dictionary browser. This simple DEB client demonstrates several basic functions of the system:
  - multilingual user interface (English, Czech, other can be easily added)
  - queries to several XML dictionaries (with different underlying structure) with the result displayed with the use of XSLT transformations
  - connection to a morphological analyzer
  - connection to an external website (Google, Answers.com)
  - connection to a geographical information system (display of geographical data directly on their positions within a cartographic map)
- Czech Onomastic Dictionary – newly prepared dictionary of Czech proper names and their origins
- PRALED – new Prague Lexical Database of Czech

3 DEBVisDic Functionality

DEBVisDic is one of the clients that are built on the DEB platform. The first step of the DEBVisDic development lies in reimplementation of the VisDic wordnet editor within this platform, which is then followed by easy extending of the tool with new features for supporting the linguistic work on wordnets.

3.1 Assets of the DEB Platform

The DEB platform is based on client-server architecture, which brings along a lot of benefits. All the data are stored on the server and considerable part of functionality is also implemented on the server, while the client application can be very lightweight.

This approach provides very good tools for team cooperation; data modifications are immediately seen by all the users. Server also provides authentication and authorization tools.

One of the main benefits of developing DEBVisDic on the DEB platform is the homogeneity of the data structure and presentation. If the wordnet administrator 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.

Of course, a drawback of the client-server architecture is that an operating server is necessary for a fully functional application. However, in special situations, the server can be installed within a local environment, or for the possibility of simple offline wordnet editing, the client may work in a degraded manner without the instant connection to the server.

3.2 Functionality of DEBVisDic

DEBVisDic uses new versatile interface (see the Figure 1 that allows the user to arrange the work without any limitations. Of course, DEBVisDic contains all the main features that were present in VisDic:

- multiple views of multiple wordnets
- freely defined text views
- synset editing
- hypero-hyponymic tree
- query result lists
- plain XML view of a synset
- synchronization
- inter-dictionary linking
- tree browsing
- consistency checks
- journaling
- user configuration

With the help of the DEB platform reusability, DEBVisDic will be supplemented with many new features that are currently accessible only as separate tools or resources. This functionality includes:

- connection to a morphological analyzer (for languages, where it is available)
- connection to language corpora, including Word Sketches statistics
- access to any electronic dictionaries stored within the DEB server
- searching for literals within encyclopedic web sites
- and many others

4 Conclusions and Future Directions

We have described the DEB implementation platform and the main features of the DEBVisDic, the successor of VisDic. The DEB platform consequently uses the client/server architecture and offers several different clients allowing to perform various lexicographic tasks. The relevant features of the DEB platform are high modularity and configurability. Thanks to them, the DEB platform represents a versatile base, on which the individual and powerful dictionary writing tools (clients) can be implemented.

VisDic, during its not so long history, proved its usefulness and contributed to the wordnet-like databases creation especially within Balkanet project. We have shown that DEBVisDic as its successor retains its functionality and adds new functions that will allow the lexicographers and researchers to create new high quality lexical resources without which further progress in the NLP field can hardly take place.

In the close future we will release DEBVisDic for the testing, and in particular, we are going to cooperate in this respect with the Tuebingen University (C. Kunze and L. Lemnitzer, Germanet)

The development of DEBVisDic is also related to current Semantic Web projects, in particular we will use the tool for building ontologies covering various domains (one of the candidates is oncology). In this connection we have to mention the tool VisualBrowser [Zuzana, 2005] developed at the NLP Laboratory FI MU which converts wordnet-like databases stored in DEBVisDic XML data format into RDF notation for natural presentation of the semantic network.

Acknowledgements

This work has been partly supported by the Czech Academy of Sciences under the projects 1ET100300414 and 1ET100300419 and by Ministry of Education CR under the project LC536.

References

Balkanet project website, `http://www.ceid.upatras.gr/Balkanet/`.

L. Benítez, S. Cervell, G. Escudero, M. López, G. Rigau, and M. Taulé. 1998. Methods and tools for building the Catalan WordNet. In *workshop on Language Resources for*

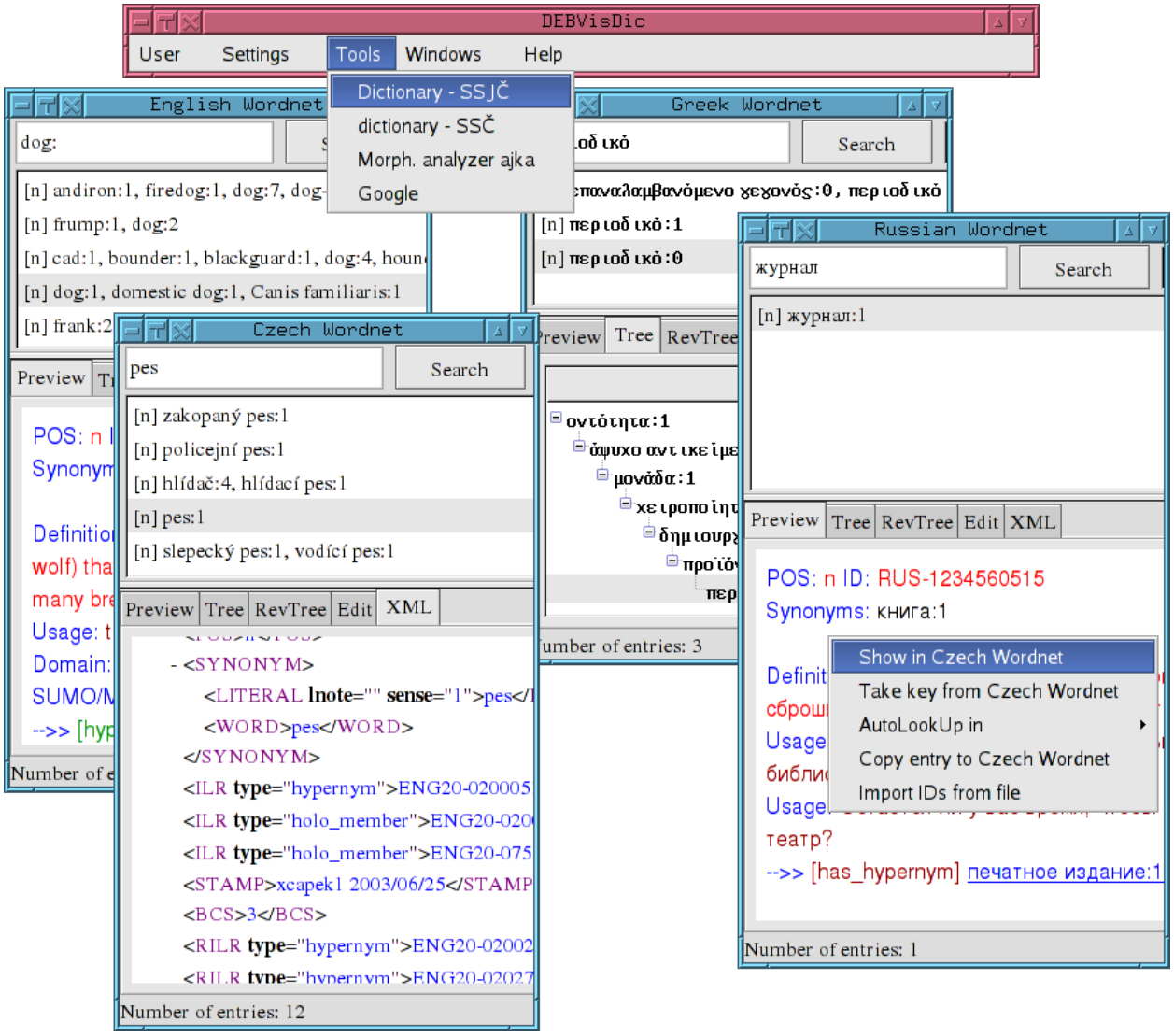

Figure 1: The new DEBVisDic main interface

European Minority Languages on LREC 1998, Granada, Spain.

Eurowordnet project website, `http://www.illc.uva.nl/EuroWordNet/`.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, Massachusetts, London, England.

Aleš Horák and Pavel Smrž. 2003. VisDic – wordnet browsing and editing tool. In *Proceedings of the Second International WordNet Conference – GWC 2004*, pages 136–141, Brno, Czech Republic.

Aleš Horák and Pavel Smrž. 2004. New features of wordnet editor VisDic. In *Romanian Journal of Information Science and Technology*, volume 7, pages 1–13.

Louw M. 1998. Polaris user's guide. Technical report, Belgium.

Ian et al. Oeschger. 2002. *Creating Applications with Mozilla*. O'Reilly and Associates, Inc., Sebastopol, California.

Louis Rosenfeld and Peter Morville. 1998. *Information Architecture for the World Wide Web*. O'Reilly and Associates, Inc., Sebastopol, California.

Pavel Smrž and Martin Povolný. 2003. Deb - dictionary editing and browsing. In *Proceedings of the EACL03 Workshop on Language Technology and the Semantic Web: The 3rd Workshop on NLP and XML (NLPXML-2003)*, pages 49–55, Budapest, Hungary.

SOAP 1.1 – W3C Simple Object Access Protocol (SOAP) 1.1 Specification, `http://www.w3.org/TR/SOAP/`.

Nevěřilová Zuzana. 2005. Visual browser, `http://nlp.fi.muni.cz/projects/visualbrowser/`.