DEBWrite: Free Customizable Web-based Dictionary Writing System

Adam Rambousek, Aleš Horák

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{rambousek,hales}@fi.muni.cz

Abstract

Today, lexicographers can avail themselves of several commercial and freely distributed dictionary writing systems (DWS). Nevertheless, there is still a group of users whose requirements are not satisfied by existing DWSs. In various lexicographic forums, there is a growing demand for freely available DWS that allows customization of the dictionary microstructure. In accordance with such requests, a new project was developed as part of the DEB (Dictionary Editor and Browser) platform. DEBWrite is implemented as a multi-platform web application based on open standards. It allows users to create and share a new dictionary without any difficult configuration or advanced technical skills. According to a defined entry structure, the editing form and the public dictionary browser are generated automatically. DEBWrite supports small and larger team cooperation when working on the dictionary content. Access rights management for the created dictionary involves three levels of user roles: a manager, an editor, and a reader. It is possible to publish the resulting dictionary in various formats, both for human readers, and for external applications (e.g. NLP-related applications that need to work with lexicographic data). The dictionary may be published in an online form, or in formats suitable for print preparation.

Keywords: dictionary writing system; lexicographic platform; dictionary authoring; DEB platform

1. Introduction

There are several software tools available for dictionary creation and publication, both commercial (e.g. IDM DPS (IDM DPS, 2006) or TLex (Joffe and de Schryver, 2004)), and freely available (e.g. Mātāpuna (Moskovitz, 2004)). During the development of the DEB (Dictionary Editor and Browser) lexicographic platform (Horák and Rambousek, 2007; Horák et al., 2008), we have designed and implemented many lexicographic projects with complex entry structure or management. On the other hand, we have also experienced demand for dictionary writing software in the form of small size dictionaries with entry structure, usually by a small lexicographic team with limited resources for their project. For such teams, existing free tools are too limiting, and commercial tools are too expensive. Several such dictionaries were created using the DEB platform tools. For example, the Terminological Dictionary of Fine Arts by the Faculty of Fine Arts, Brno University of Technology (Horák and Rambousek, 2007), or the Czech-English Dictionary of Ethnological

Terminology by the The National Institute of Folk Culture[1]. To fulfil the requirements for such range of dictionaries, a new application of the DEB platform was developed, called DEBWrite.

2. The DEB platform

Utilizing the experience from several preceding lexicographic projects, we have designed and implemented a universal dictionary writing system that can be exploited in various lexicographic applications to build distributed lexical databases. The system is called Dictionary Editor and Browser, or the DEB platform (Horák and Rambousek, 2007, 2010). Since 2005, the DEB platform was applied in more than 10 large international research projects. Large-scale applications based on the DEB platform include the lexicographic workstation for the development of the Czech Lexical Database (Horák and Rambousek, 2013) with detailed morpho-syntactic information on more than 213,000 Czech words, or the complex lexical database Cornetto combining the Dutch wordnet, an ontology, and an elaborate lexicon (Horák et al., 2008). Currently ongoing projects include Pattern Dictionary of English Verbs tightly interlinked with the corpus evidence (Maarouf et al., 2014), Family names in Britain and Ireland (Hanks et al., 2011) providing detailed investigations for over 45,000 surnames to be published by Oxford University Press, or the dictionary of the Czech Sign Language[2] with an extensive use of video recordings to present the signs (Rambousek and Horák, 2015).

The DEB platform is based on the client-server architecture, which brings along a lot of benefits. All the dictionary and interlinked data are stored on a server and a considerable part of the functionality is also implemented on the server-side, consequently the client application can be very lightweight. This approach provides very good tools for editor team cooperation; data modifications are immediately seen by all involved users. The DEB server also provides authentication and authorization tools.

The server part is built from small, reusable parts, called servlets, which allow a modular composition of all services. Each servlet provides different functionality such as database access, dictionary search, morphological analysis or a connection to corpora. The overall design of the DEB platform focuses on modularity. The data stored in a DEB server can use any kind of structural database (or consult several databases and join them into one compact dictionary storage) and prepare and combine complex results of answers to user queries without the need to use specific query languages for each data source. The main data storage is currently provided by the Sedna XML database (Fomichev et al., 2006), which is an open-source native XML database providing XPath and XQuery access to a set of document containers. Several DEB applications also work with connections to standard relational databases, such as PostreSQL or MySQL, or to specialized data providers, such as the geographical information system GRASS or a morphological analyser.

[1] http://www.nulk.cz

[2] http://www.dictio.info

The user interface, which forms the most important part of a client application, usually consists of a set of flexible complex forms that dynamically cooperate with the server parts. Client applications can be implemented in any programming language that allows to interact with the DEB server using the available server interfaces.

Client applications communicate with servlets using standard HTTP requests in a manner similar to a popular concept in web development called AJAX (Asynchronous JavaScript and XML) or using the SOAP protocol[3]. The data are transported over HTTP in a variety of formats – RDF, XML documents, JSON-encoded data[4], plain-text formats, or marshalled using SOAP.

The main assets of the DEB development platform can be characterized by the following points:

- All the data are stored on the server and a considerable part of the functionality is also implemented on the server, while the client application can be very lightweight.
- Very good tools for (remote) team cooperation; data modifications are immediately seen by all the users. The server also provides authentication and authorization tools.
- Server may offer different interfaces using the same data structure. These interfaces can be reused by many client applications.
- Homogeneity of the data structure and presentation. If an administrator commits a change in the data presentation, this change will automatically appear in every instance of the client software.
- Integration with external applications.

2.1 Linked Data

The term Linked Data refers to a methodology for publishing and interlinking structured data online. This methodology was proposed by Berners-Lee in 2006 (Berners-Lee, 2006; Bizer et al., 2009), who outlined four rules of how data are required to meet for easy sharing and interconnecting:

1. objects are identified by an URI[5] (e.g. `http://dbpedia.org/page/Brno`),
2. URI identifiers are HTTP links, where people or software tools can access the data,
3. useful information are provided on given URI, using the appropriate standards (like RDF) (the previously mentioned page contains links to the same information in multiple formats, RDF is provided at `http://dbpedia.org/data/Brno.rdf`),
4. other objects are referenced using their URIs to get more information (e.g. link from the `Brno.rdf` to `http://dbpedia.org/resource/South_Moravian_Region`).

All resources stored in the DEB platform can be published using the Linked Data methodology. The DEB platform provides the tools for Linked Data presentation and the decision how to release the data lies with the author. Linked Data requirements are satisfied in the following manner:

[3] `http://www.w3.org/TR/2007/REC-soap12-part0-20070427/`
[4] `http://www.json.org/xml.html`
[5] Uniform resource identifier (Berners-Lee et al., 2005)

1. use URIs as names – each entry has a unique URI identifier,
2. use HTTP URIs – through the DEB platform API, entries are accessible on HTTP URI,
3. provide useful information using standards – when linking to an entry URI, the data are displayed either in raw XML format, or converted to RDF or other defined format,
4. link to other URIs – the DEB platform enables to link to other resources if provided by the data author.

These requirements are fully embraced in DEB-based projects, DEBVisDic (Horák et al., 2006) and the KYOTO project (Horák and Rambousek, 2010, 2009), where all the information were released as Linked Data.

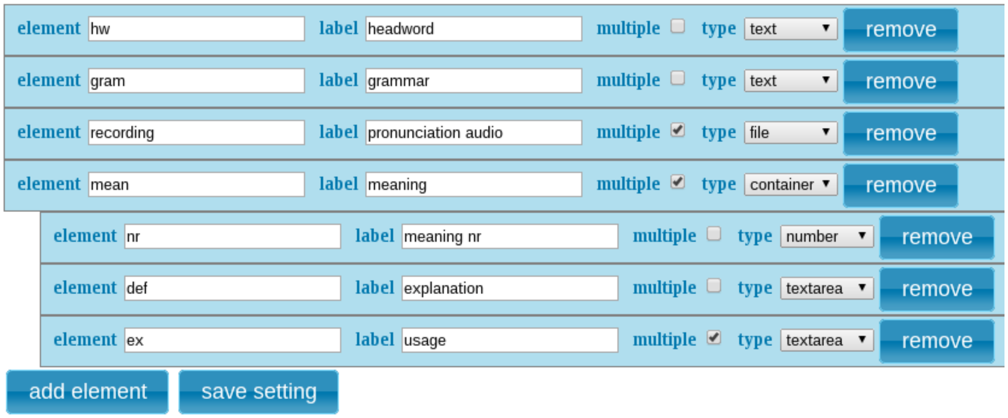
Berners-Lee later published a rating system for the distributed data, while expanding the term Linked Data to Linked Open Data – which means Linked Data that are released under an open licence. This rating system is aimed especially at government agencies to encourage them to publish valuable (and reusable) information. The importance of Linked Open Data is acknowledged for example by the European Union, funding projects like *LOD2* (large integrating project to develop tools, standards and management methods for Linked Open Data) or *Open Data Portal* (catalogue of data available for reuse). The rating system follows these principles:

- 1 star – the data are available on the web in any format, with an open licence.
- 2 stars – the data are published in machine-readable structured format.
- 3 stars – the data use non-proprietary format.
- 4 stars – W3C open standards (RDF and SPARQL) are used to identify objects for linking.
- 5 stars – the data contain links to other resources to give context.

The DEB platform offers a full support to the dictionary publisher to disseminate the dictionary content as Linked Open Data:

1. published online with an open licence – this has to be decided by the data authors, but the DEB platform enables releasing data on the web.
2. available as machine-readable structured data – documents in the DEB platform are stored in an XML format which is machine-readable.
3. non-proprietary format – XML is a standardized format.
4. use open standards from W3C (RDF and SPARQL) – XML format itself is the W3C standard, but to conform with this requirement more precisely, documents are converted to RDF format.
5. link to other resources – the DEB platform enables interlinking to other resources.

As demonstrated, the only limitation is the decision of the data authors regarding the licensing. When this is resolved, the DEB platform enables to publish all documents as Linked Open Data.



element	hw	label	headword	multiple	□	type	text ▾	remove
element	gram	label	grammar	multiple	□	type	text ▾	remove
element	recording	label	pronunciation audio	multiple	☑	type	file ▾	remove
element	mean	label	meaning	multiple	☑	type	container ▾	remove
element	nr	label	meaning nr	multiple	□	type	number ▾	remove
element	def	label	explanation	multiple	□	type	textarea ▾	remove
element	ex	label	usage	multiple	☑	type	textarea ▾	remove

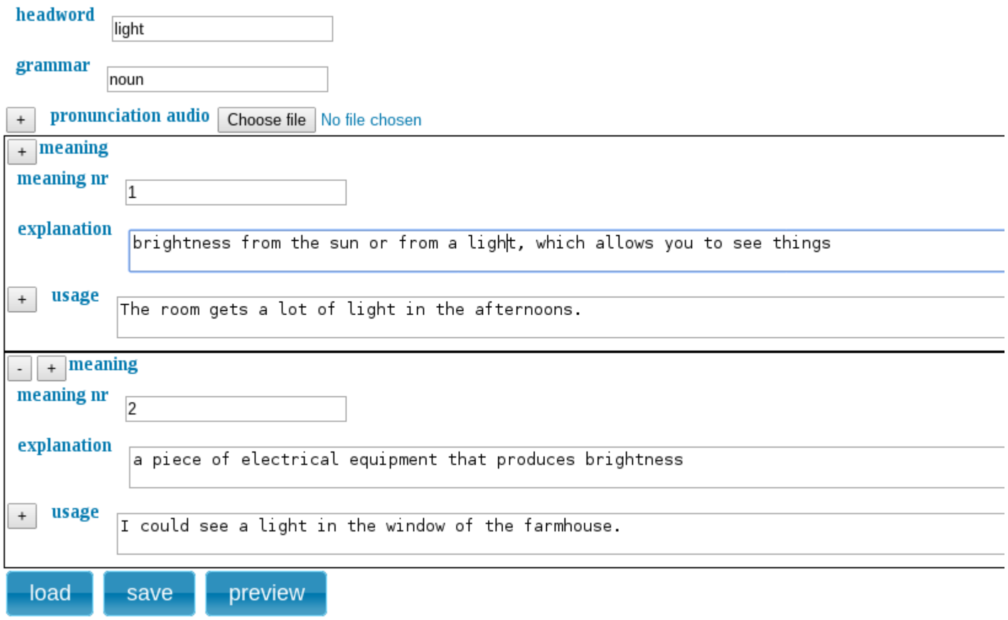
Figure 1: Setting the entry structure.

3. The DEBWrite application

The DEBWrite application is implemented as a multi-platform web application, utilizing HTML5 and JavaScript standards[6] that allow full interoperability and dynamic adaptations to current dictionary interfaces. The DEBWrite application allows users to create and share a new dictionary without any complicated configuration or advanced technical skills. Based on experience with dictionaries in the DEB platform, a default entry structure is proposed that fits many dictionaries (also with terminological dictionaries in mind). Each entry is composed of a top level information (headword and its variants, grammatical information, domain/category) and any number of meanings (each containing explanation and usage examples). Translations to various languages, cross-references to other entries (with relation type), collocations, and external references may be included on the entry level or meaning level. Within the dictionary definition form, users may alter the entry structure in a graphical interface (see Figure 1) – deleting unnecessary information or adding new entry fields, changing labels, or altering the option lists (relation types, languages for translations, domains...).

According to the updated entry structure, the editing form and the public browser are generated automatically. See Figure 2 for an example of the editing form. The dictionary website design is fully customizable via CSS stylesheets or templates that are used for output generation. XSLT templates are used as a default option, however HandlebarsJS template engine[7] is also evaluated. Based on the user feedback, the preferred template engine might be changed in the future DEBWrite updates. The authors may either edit the source code of the output generating files, or select some of the variables (e.g. colours and font styles) in the graphical interface (see Figure 3). In future versions, more detailed graphical interface to change the output layout will be added. Each dictionary may use multiple output templates to provide different dictionary previews based on user settings.

[6] with jQuery, `https://jquery.com/`, and jQuery UI, `https://jqueryui.com/`, libraries.
[7] `http://handlebarsjs.com/`



headword: light

grammar: noun

pronunciation audio: No file chosen

meaning

meaning nr: 1

explanation: brightness from the sun or from a light, which allows you to see things

usage: The room gets a lot of light in the afternoons.

meaning

meaning nr: 2

explanation: a piece of electrical equipment that produces brightness

usage: I could see a light in the window of the farmhouse.

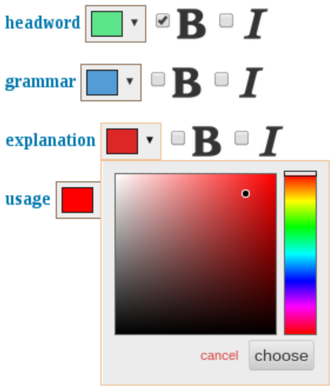
Figure 2: Example of the editing form automatically generated from the settings.

The DEBWrite dictionary editor also supports upload of multimedia attachments (e.g. large figures, audio or video recordings) to supplement the entries. The authors need to specify a special field type in the entry structure for file uploads. The server detects the attachment type (e.g. image, video, audio) and displays the multimedia content in an appropriate form for the output. See Figure 4 for an example of multimedia file upload and output.

In cases, when the lexicographers have some information prepared in advance, DEBWrite can simplify the start of the dictionary creation process. A common scenario includes the situation, where DEBWrite imports a list of headwords and automatically creates corresponding empty entries prepared for expert editing. Another scenario works with the requirement of moving rich existing structured data to DEBWrite. In such cases, DEBWrite can import a (part of the) full dictionary in the XML format. As of now, the imported file must follow the XML structure used in the DEBWrite application internally. However, a conversion between different (compatible) XML structures is a matter of applying an XSLT template conversion. Future versions of DEBWrite will support also import of data in custom XML format.

The application also supports an export to standard XML file. Preprocessed XSLT templates are included to export converted dictionary data into an HTML format for online publishing. For printed or electronic edition in PDF, the data are converted to $\LaTeX$ and subsequently to PDF format.

To enhance the possibility to share and re-use lexicographic resource sharing, DEBWrite also provides the data in the form compliant with the Linked Data methodology (see section 2.1). The decision about the data licensing and access control lies entirely on the dictionary authors, however DEBWrite provides the tools needed to make the sharing easy.



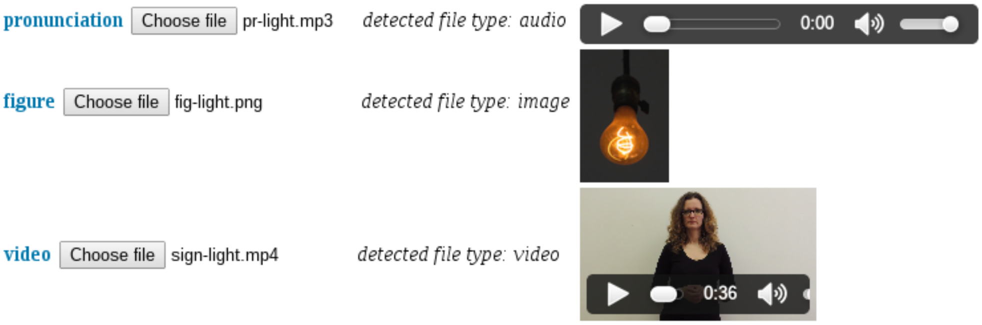
headword 🟩 ☑ **B** □ *I*

grammar 🟦 □ **B** □ *I*

explanation 🟥 □ **B** □ *I*

usage 🟥

Figure 3: Example of output design customizations.



pronunciation Choose file pr-light.mp3 *detected file type: audio*

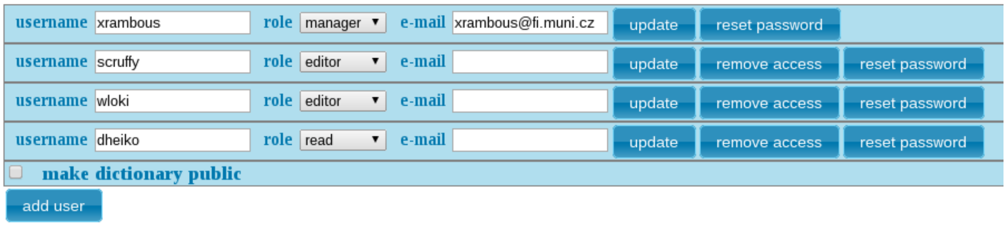
figure Choose file fig-light.png *detected file type: image*

video Choose file sign-light.mp4 *detected file type: video*

Figure 4: Output representation of various media attachment types.

One of the major advantages of the DEBWrite application lies in its support of a team cooperation on the dictionary preparation process. DEBWrite classifies authorized users into one of three possible user roles: a manager, an editor, or a reader (see Figure 5 for example of user access management).

- The user who created the dictionary is the dictionary *manager*. Managers may alter any dictionary settings. They may grant access to the dictionary to other users, specifying their role. Managers are able to edit all the dictionary entries and set an entry for publication. The manager may also decide to make published entries publicly available, which means that no password is needed to browse the dictionary (this might be regarded as a fourth user role in the dictionary access management).
- An *editor* may edit entries before they are set to be published.
- *Readers* may browse and navigate through the published entries and their attachments with advanced search capabilities.



username xrambous role manager e-mail xrambous@fi.muni.cz
username scruffy role editor e-mail
username wloki role editor e-mail
username dheiko role read e-mail
□ make dictionary public

Figure 5: User access management.

4. Conclusions

We have introduced a new customizable and freely available dictionary writing system named DEBWrite. The application prototype is currently in public testing, available at `http://deb.fi.muni.cz/debwrite`. As a part of testing, the Terminological Dictionary of Fine Arts was converted to DEBWrite from the original application (where the editing form functionality was originally limited to the Firefox browser only), allowing multi-platform editing and providing better user experience.

5. Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013. The research leading to these results has received funding from the Norwegian Financial Mechanism 2009–2014 and the Ministry of Education, Youth and Sports under Project Contract no. MSMT-28477/2014 within the HaBiT Project 7F14047.

6. References

Berners-Lee, T. (2006). Design Issues: Linked Data.

Berners-Lee, T., Fielding, R. & Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. STD 66 (INTERNET STANDARD).

Bizer, C., Heath, T. & Berners-Lee, T. (2009). Linked Data-The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), pp. 1–22.

Fomichev, A., Grinev, M. & Kuznetsov, S. (2006). Sedna: A Native XML DBMS. *Lecture Notes in Computer Science*, 3831:272.

Hanks, P., Coates, R. & McClure, P. (2011). Methods for Studying the Origins and History of Family Names in Britain. In *Facts and Findings on Personal Names: Some European Examples*, Uppsala. Acta Academiae Regiae Scientiarum Upsaliensis, pp. 37–58.

Horák, A., Pala, K., Rambousek, A. & Povolný, M. (2006). DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool. In *Proceedings of the Third International WordNet Conference - GWC 2006*, Jeju, South Korea. Masaryk University, Brno, pp. 325–328.

Horák, A. & Rambousek, A. (2007). DEB Platform Deployment – Current Applications. In *RASLAN 2007: Recent Advances in Slavonic Natural Language Processing*, Brno, Czech Republic. Masaryk University, pp. 3–11.

Horák, A. & Rambousek, A. (2009). Using Wordnets and Ontologies for Text-Meaning Assignment - Implementation Details of the KYOTO Project First Phase. In *Proceedings of the 4th International Conference on Software and Data Technologies, Volume 2*, Portugal. INSTICC, pp. 303–307.

Horák, A. & Rambousek, A. (2010). Using DEB Services for Knowledge Representation within the KYOTO Project. In *Principles, Construction and Application of Multilingual WordNets, Proceedings of the Fifth Global WordNet Conference*, New Delhi, India. Narosa Publishing House, pp. 165–170.

Horák, A. & Rambousek, A. (2013). PRALED – A New Kind of Lexicographic Workstation. In Przepiórkowski, A., Piasecki, M., Jassem, K. & Fuglewicz, P., editors, *Computational Linguistics: Applications*, Springer, pp. 131–141.

Horák, A., Vossen, P. & Rambousek, A. (2008). A Distributed Database System for Developing Ontological and Lexical Resources in Harmony. In *Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing*, Haifa, Israel. Springer-Verlag, pp. 1–15.

IDM DPS (2006). IDM Dictionary Production System. http://www.idm.fr/products/dictionary_writing_system.

Joffe, D. & de Schryver, G.-M. (2004). TshwaneLex – Professional off-the-shelf lexicography software. In *Third International Workshop on Dictionary Writing Systems: Program and List of Accepted Abstracts*, Brno, Czech Republic. Masaryk University, Faculty of Informatics. http://tshwanedje.com/tshwanelex/.

Maarouf, I. E., Bradbury, J., Baisa, V. & Hanks, P. (2014). Disambiguating verbs by collocation: Corpus lexicography meets natural language processing. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. & Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Moskovitz, D. (2004). Mātāpuna Dictionary Database System. In *Third International Workshop on Dictionary Writing Systems: Program and List of Accepted Abstracts*, Brno, Czech Republic. Masaryk University, Faculty of Informatics. http://matapuna.thinktank.co.nz/matapuna/.

Rambousek, A. & Horák, A. (2015). Management and Publishing of Multimedia Dictionary of the Czech Sign Language. In Biemann, C., Handschuh, S., Freitas, A., Meziane, F. & Métais, E., editors, *Natural Language Processing and Information Systems, NLDB 2015*, Lecture Notes in Computer Science, Springer, pp. 399–403.