New Technology Platform for the Multilingual Sign Language Dictionary

Adam Rambousek

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

Abstract. Since 2014, Teiresiás Centre at Masaryk University is co-ordinating the project to create the multilingual sign language dictionary. Natural Language Processing Centre is developing the editing and browsing web application for the dictionary. Originally, the application was based on the DEB dictionary platform with Sedna XML database for data storage. In course of the project, more languages were added, entry structure is more complex, larger teams from several countries are working on the dictionary creation, and website design was not working very well with modern web browsers. We realized that in order to increase the response speed of the application we need to refactor the whole technology platform. In 2020 and 2021, completely new application was designed and developed. This paper this describes the overall structure of the platform, technologies used to build the application and the process of data migration to the new database system.

Keywords: Dictionary editing · Dictionary writing system · Sign language · XML · JSON · MongoDB database

1 Introduction

In 2014, the Teiresiás Centre at Masaryk University was co-ordinating the project which aimed to build the Czech Sign Language dictionary connected with the Czech dictionary. Several organizations were working on the dictionary data, and the Natural Language Processing Centre was asked to develop web application to view and edit dictionary entries. Application was built using the DEB platform tools [9,5] – data were encoded in the XML format and stored in the Sedna XML database, for editing custom web editor was developed in Javascript, for viewing entries were converted from the XML format to HTML using XSLT templates. More details about the application are described in [8].

1.1 Languages and Entry Structure

Over the years, more international organizations joined the project and thus more languages were added. Dictionary application is called *Dictio – Multilin-*

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2021, pp. 3–10, 2021.

gual dictionary focused on sign languages[1]. Currently, the dictionary contains the following languages:

- Czech Sign Language (Český znakový jazyk, ČZJ),
- Slovak Sign Language (Slovenský posunkový jazyk, SPJ),
- Austrian Sign Language (Österreichische Gebärdensprache, ÖGS),
- American Sign Language (ASL),
- International Signs (IS),
- Czech,
- Slovak,
- German,
- English.

General entry structure is the same for all languages, however level of details in each part is different for various languages:

- headword,
- grammar information (at least Part-of-Speech, ideally all morphological details),
- etymology of the word or sign,
- stylistic information (regional or limited usage, etc.),
- for sign languages, transcription into SignWriting or HamNoSys [10,7],
- meanings
  - definition,
  - usage examples,
  - translations,
  - other semantic relations (e.g. synonyms, hypernyms).

Of course, the main difference between sign and spoken languages is the headword representation – headword is represented with the video recordings (front and side view) of the person showing the sign. In Dictio, unlike in other sign language dictionaries, even the definition and usage examples are presented as video recordings in sign language.

As for translations, at least the entries in sign language and its spoken counterpart (e.g. ČZJ and Czech, or ÖGS and German) are connected. But it is possible to add translations to any other language and web application supports searching in all of the language pairs.

Currently, Dictio contains 158,357 entries and 70,501 videos altogether, see Table 1 for details about the number of entries and recordings in each language.

2 Technology

After evaluation of tools used in the first version of Dictio, it was decided to implement most parts of the application from scratch.

[1] Available at `https://www.dictio.info/`.

Table 1: Number of entries and video recordings per language

language	entries	videos
Czech	120,274	
Czech Sign Language	12,526	44,330
German	5,652	
Slovak	5,590	
English	5,555	
Slovak Sign Language	4,812	17,300
Austrian Sign Language	3,436	7,400
International Signs	369	1,050
American Sign Language	127	290

Main part of the application logic was implemented in Ruby programming language[2] [3]. Some complex underlying functions may be kept with just a small updates, e.g. combining the SignWriting signs for collocation entries, or processing inter-language relations changes. For that reason, we decided to implement new application in Ruby, but updating the code from Ruby 1.8 to Ruby 2.6. Apart from keeping with current development, this update also introduced better handling of UTF-8 strings. Thus, all the tools and libraries used in the new application need to support Ruby.

2.1 Database

Entry structure is very complex and while it is stable after the development of the first version, there might still be structure changes in the future. Originally, entries were saved in the XML format and stored in Sedna XML database [4]. We needed to either keep the XML format, or use format with the same complexity.

With growing number of entries and links between them, the performance of Sedna XML database was getting worse. Unfortunately, Sedna is no longer actively developed, thus we had to select another database. We evaluated performance benchmarks for open-source XML and NoSQL databases. We decided to use MongoDB NoSQL database[3] [1,6].

MongoDB stores documents in the JSON format [2], or more specifically in the BSON ("Binary JSON") format[4]. BSON format is a binary representation of the JSON documents with support for more complex data types, and was designed to be more efficient both for the storage space, and the reading speed.

Because of the document format change, all the entries and metadata in the database had to be converted. This also proved to be good opportunity to clean the entry structure. We removed unnecessary nesting of data where possible to make the structure more readable. In the Sedna database, some values were

[2] `https://www.ruby-lang.org/`

[3] `https://www.mongodb.com/try/download/community`

[4] BSON format specification is available at `https://bsonspec.org/`. JSON and BSON formats are compared at `https://www.mongodb.com/json-and-bson`.

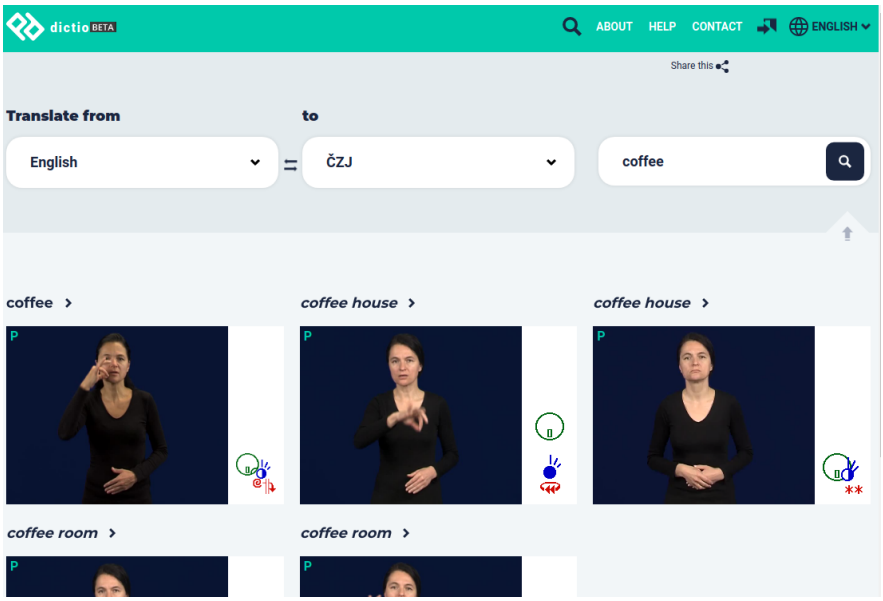

Fig. 1: Translations from English to Czech Sign language.

duplicated (e.g. information about the target entry of translation link) to speed up querying and displaying. This is not needed anymore and each information is stored only once. Originally, each language had separate database collection for entries and for video metadata. In MongoDB, all entries are stored in single database with additional language attribute (similarly for video metadata).

On the backend side, no big changes were needed, because even in the first version all the XML data were converted to objects before using them in the application. This is much easier with BSON documents provided by the MongoDB API.

On the frontend side, the editor for creating and updating the entries had to be updated. The application is implemented in JavaScript and provides complex editing form for users. Fortunately, we had to update just the two functions: to load the XML document from the database and parse the data to form boxes, and to get the form data and send the XML document to the database. Obviously, these functions were re-implemented to work with the documents in JSON format.

2.2 Web Application Tools

Original version of Dictio used the Webrick server to process network requests and a set of custom templates and XSL stylesheets to display the web pages. Main disadvantages of the Webrick server are the worse performance with high load of requests, and support for only single-threaded processing.

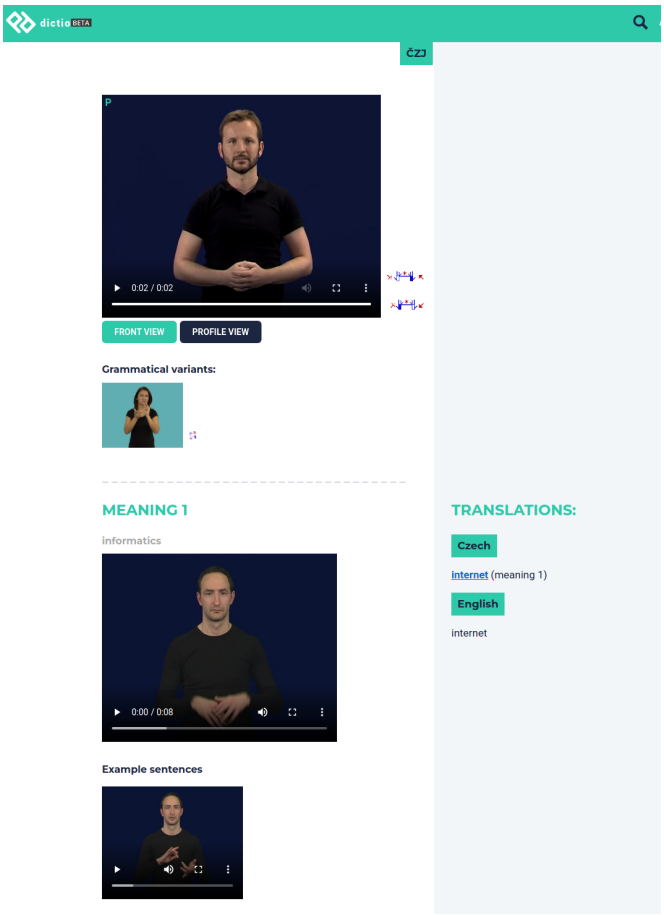

Fig. 2: Full details of single Czech Sign Language entry.

After performance evaluation of existing tools, we decided to use the Sinatra framework[5] for creating web applications. Sinatra is used for request processing, routing, user authentication, user session setting and application interface.

To display web pages with the data to the users, we selected the Slim template engine[6] Templates in Slim contain as little HTML formatting as possible, document structure is based on template indentation, and main focus in template writing is on the data. It is also possible to re-use and combine templates, which is advantage for well arranged implementation. Completely new web page design was created with support for mobile devices. See Figure 1 for example of result for translation search from English to Czech Sign Language. Figure 2 shows an example with full information about single entry in Czech Sign Language. See Figure 3 for example of layout for mobile devices, with results of translation from English to Czech Sign Language.

[5] Available at `http://sinatrarb.com/`.
[6] Available at `http://slim-lang.com/`.

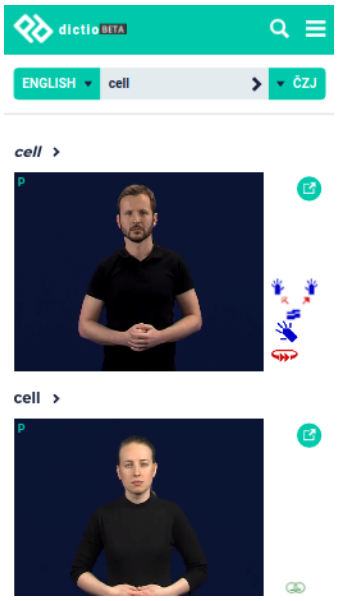

Fig. 3: Responsive design for mobile devices.

3 Platform Structure

In the Dictio project, there are several groups of users of the application with various needs:

- general public – browsing and querying the entries,
- editors – adding or updating the entries, uploading video recording, based on the department they belong to,
- dictionary managers – reviewing entries, assigning work based on reports about missing parts of entries, managing users and their access permissions.

In the original Dictio version, all users were working on the same server. Also the database and all the video files were stored on the same machine. This arrangement had bad impact on the overall performance and user experience. For example, when many users were browsing the dictionary, the entry editing application was responding slower. Similarly, when mass import of video recording was under way, users were waiting too long for entry display.

To improve the application performance and also to keep different tasks separate, we designed new platform structure. Application is now split into five independent virtual servers, provided by the MetaCentrum Cloud[7]:

- database server with MongoDB,
- file server with all the video recordings (`files.dictio.info`),
- public viewing server (`www.dictio.info`),
- editing server (`edit.dictio.info`),
- administration server (`admin.dictio.info`).

[7] `https://cloud.muni.cz/`

All three web servers (www, edit, admin) share the same source code and thanks to Sinatra conditional routing only the appropriate parts and templates are provided. Database server is accessible only via internal network from the web servers, and is not open to public network.

4 Conclusion and Future Developments

We re-implemented the Dictio multilingual sign language dictionary as completely new web application. We decided to change the database, document storage format, web framework, and template engine. Using current technology and more modular application structure is providing better performance and better experience for users. Currently, all functionality of the original application is supported. New application is in regular use since March 2021 and we are continuously adding new features based on user feedback.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Chodorow, K., Dirolf, M.: MongoDB: The Definitive Guide. O'Reilly Media, Inc., 1st edn. (2010)
2. Crockford, D.: JSON, The Fat-Free Alternative to XML. In: Proceedings of XML 2006. Boston, USA (2006), http://www.json.org/xml.html
3. Flanagan, D., Matsumoto, Y.: The Ruby Programming Language. O'Reilly, first edn. (2008)
4. Fomichev, A., Grinev, M., Kuznetsov, S.: Sedna: A Native XML DBMS. Lecture Notes in Computer Science **3831**, 272 (2006)
5. Horák, A., Rambousek, A.: Lexicographic tools to build new encyclopaedia of the czech language. The Prague Bulletin of Mathematical Linguistics **2016** (2016). https://doi.org/http://dx.doi.org/10.1515/pralin-2016-0019, https://ufal.mff.cuni.cz/pbml/106/art-horak-rambousek.pdf
6. Jose, B., Abraham, S.: Performance analysis of nosql and relational databases with mongodb and mysql. Materials Today: Proceedings **24**, 2036–2043 (2020). https://doi.org/https://doi.org/10.1016/j.matpr.2020.03.634, https://www.sciencedirect.com/science/article/pii/S2214785320324159
7. Kato, M.: A Study of Notation and Sign Writing Systems for the Deaf. Intercultural Communication Studies **17**(4), 97–114 (2008)
8. Rambousek, A., Horák, A.: Management and Publishing of Multimedia Dictionary of the Czech Sign Language. In: Biemann, C., Handschuh, S., Freitas, A., Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems, NLDB 2015. pp. 399–403. Lecture Notes in Computer Science, Springer (2015). https://doi.org/10.1007/978-3-319-19581-0_37

9. Rambousek, A., Horák, A., Parkin, H.: Software tools for big data resources in family names dictionaries. Names **66** (2018). https://doi.org/http://dx.doi.org/10.1080/00277738.2018.1453276, https://www.tandfonline.com/doi/full/10.1080/00277738.2018.1453276
10. Sutton, V.: SignWriting Basics. Center for Sutton Movement Writing, Incorporated (2009)