SHAWEL: Sharable and interactive Web-Lexicons

Greg Gulrajani

Max-Planck-Institute for Psycholinguistics

greg.gulrajani@mpi.nl

Abstract

A prototypical lexicon tool was implemented which was intended to allow researchers to collaboratively create lexicons of endangered languages. Increasingly often researchers documenting or analyzing a language work at different locations. Lexicons that evolve through continuous interaction between the collaborators can only be efficiently produced when it can be accessed and manipulated via the Internet. The SHAWEL tool was developed to address these needs; it makes use of a thin Java client and a central database solution.

1. Introduction

Lexicons are very important for the documentation of endangered languages since they contain information centered on words and concepts along the known linguistic dimensions such as orthography, morphology, syntax, and semantics. Therefore, lexicons are one of the major data types in the DOBES program [1]. They generally appear as wordlists or as more complex dictionaries. A recent investigation [2] has shown that dictionaries used to document endangered languages differ largely in structure and linguistic content dependent on the languages and on the researchers objectives. For the purpose of the work reported in this paper a lexicon with a simple table type structure was chosen.

There is a strong need to share lexicons on the Internet between collaborators working in different locations and with the interested user community in the field. The creation of a dictionary often is a collaborative effort and is subject of inceptions in all respects. Existing entries are discussed and continuously changed, records are added and often the structure of a dictionary is modified. Until now, spreadsheets, tables or even text document files are sent back and forth between the researchers to accomplish this task. It was found that this method is inconvenient, time consuming, and prone to errors.

Another highly important aspect of material about endangered languages is how to make it usable by the indigenous communities. Here paper material or special CDROMs are the most appropriate forms. However, some communities have started using the Internet and may want to use such online dictionaries and perhaps add comments to it. For that purpose, the design of the user interface is crucially important. C. Manning and his colleagues [3] worked on a very interesting visualization and exploration option that allows indigenous people to operate in a semantic space. With simple methods, they can explore the different semantic relations between words and concepts. This method is very promising, but was not yet included in SHAWEL. Nevertheless, simplicity was one of the major goals for SHAWEL.

Therefore, in the DOBES program [4] the requirement came up to test web-based methods for creating lexicons. This paper reports a pilot program that is already used by one of the linguistic teams in the program.

2. Goals

For the first version, a number of goals were defined. Later, when the concept turns out to be successful other features will be included. They are described under the future perspectives section.

As indicated, the online dictionary should be usable even by members of the indigenous communities or other naive computer users. Therefore, simplicity of the user interface was one of the major design criteria. It was decided that a spreadsheet-like presentation is preferable[1].

The environment should allow specific users to edit fields in a multi-user environment including record locking and transaction support. All other users should be able to read it. Fine grained user permissions (administrator, read/edit/delete/ create/create accounts) were seen as important feature as well.

The setup should allow various researchers to import their lexicons, i.e. there could be dictionaries for different languages, but also different versions of lexicons. The user interface should allow the user to easily access the needed lexicon. To avoid forcing the user to select a lexicon when starting the tool from a dedicated web-site (about a certain language for example) there should be a transparent launch of the corresponding dictionary without any user id and with read permission as the default.

A multilevel UNDO option was seen as necessary to keep maximal control by the editor about what he is doing and to guarantee efficiency.

Since many teams working with different character sets and fonts should be able to use this tool, it should support the full UNICODE character range for all operations (edit/input, search, and visualization). At this moment the following character sets and fonts are supported: IPA, ISO-Latin, Cyrillic, Chinese, Hebrew, Arabic, i.e. appropriate input methods should be offered by the user interface.

To keep control of the actions of those users who are allowed to edit the lexicon, it was seen as necessary to implement an audit trail so that all modifications can be traced.

[1] For the first version a simply structured lexicon was chosen which exists in one table. Tests for web-based lexicons with more complex structures such as from CELEX [5] have been carried out.

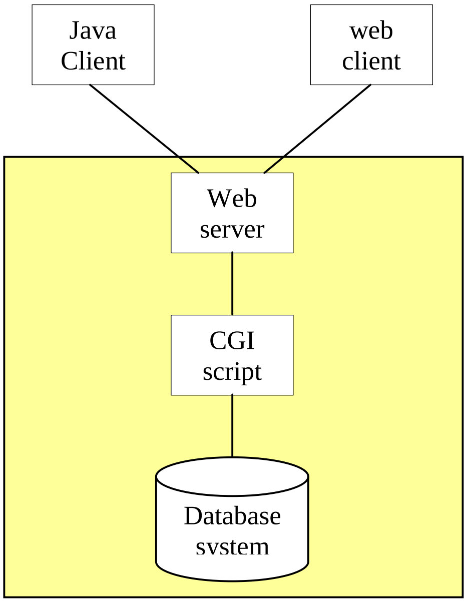
3. Design Issues

With respect to the architecture, it was decided to use a database system on a server as nucleus and to offer access via two paths: (1) A Java thin client for reading, writing, maintaining, and administrating purposes, and (2) a simple web interface for reading purposes[2]. Oracle was chosen as database server to start with[3]. It offers the following important features:

- transaction capabilities with roll-back mechanisms
- record locking mechanism
- UNICODE compatibility
- UNICODE searching and indexing support
- access right handling

The rollback function serves for a consistent database in case of incomplete transactions that could be a consequence of a broken Internet connection. Also the record locking mechanism is very important in a multi-user editing environment. Oracle offers full support for UNICODE which does not only mean that they are able to store the appropriate number of bits, but they support indexing and searching support for the many characters defined. Further, we make use of Oracle's user administration to assure that users can only do the operations they are allowed to do.

The Java based web-client is necessary for proper rendering of different character sets, and to offer a variety of input methods and to filter data to the readers. It was discussed whether people should only read lexicon data chunk-wise to prevent copying of the lexicons.

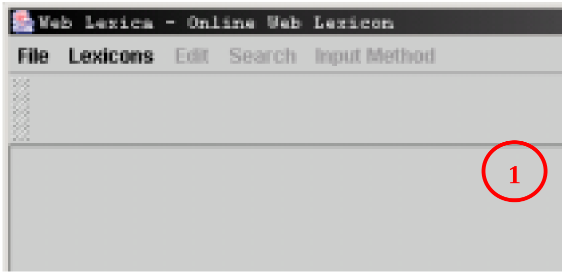


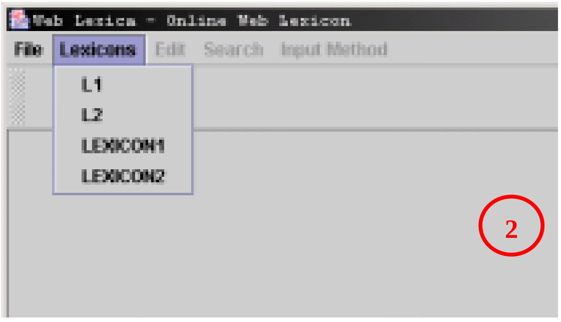
For data entry, the GUK software library [4] was used and extended[4]. The library currently offers 27 different keyboard layouts; others can be added easily. It also includes different features required by specific writing systems such as character sequence re-ordering applied in Bengali.

For downloading the thin Java client, we make use of the JNLP framework from Sun Microsystems [5]. It allows the user to launch applications from web-pages equipped with various start-up parameters such as starting a specific lexicon. Furthermore, the framework allows seamless upgrading of the applications and the ability for the application to be used without Internet connectivity.

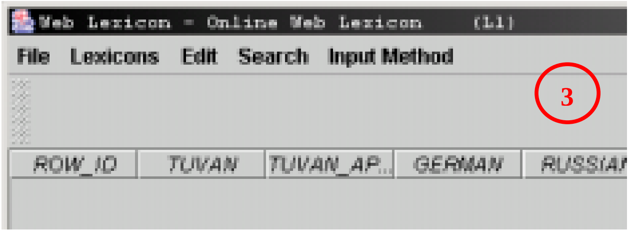
4. User Interface

When starting SHAWEL, the user will see an interface with minimal possibilities: it just shows the main menu and a menu to select one lexicon from the list of available ones. When selecting the lexicon menu the available lexicons are shown and the user can select one. For certain lexicons, the user may have to identify himself.





After having selected a lexicon, all attributes of the lexicon will be indicated. As in a spreadsheet program the user may select certain columns or rearrange the order.

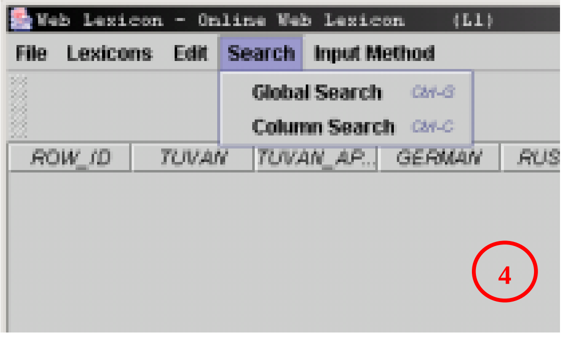


For naive users, this step may not be intuitive, since the columns at this moment are left empty. The user may want to immediately see some lexicon entries to understand the layout. The next step is to visualize lexicon information, which can be issued by querying the database. The user can decide to do a global search or a restricted search on a number of columns.

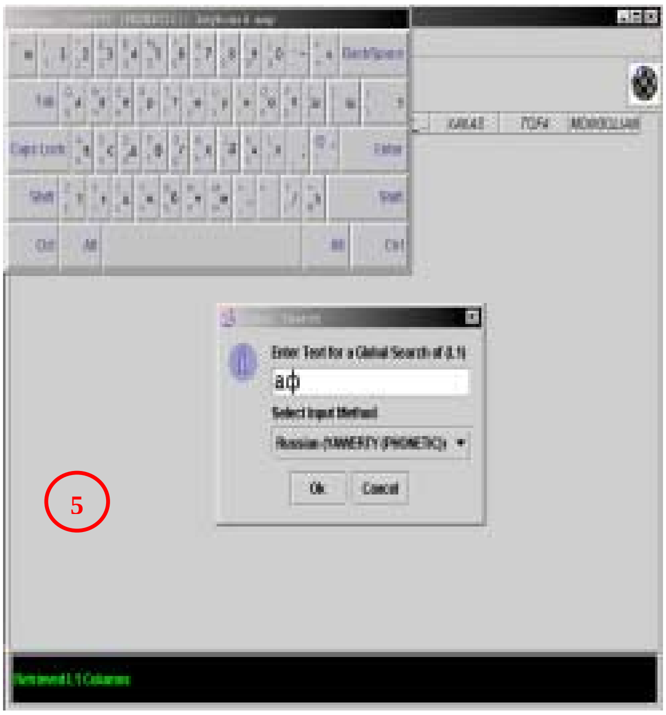
[2] This feature will be added in the second version.

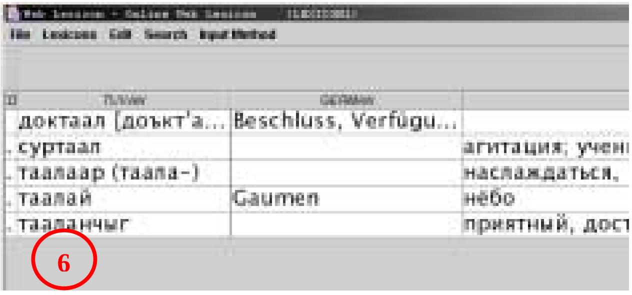
[3] Another freeware system such as mySQL may be the final choice to allow remote operation. A transformation would be relatively simple.

[4] The extensions are openly available as well.

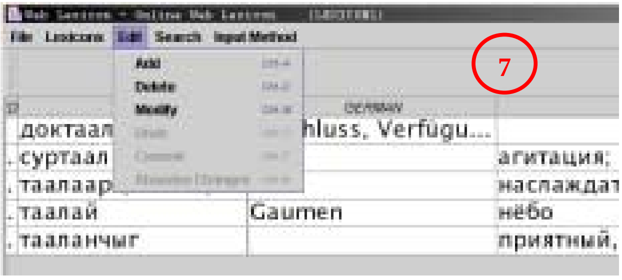


When choosing a global search, all input methods relevant for the different columns can be selected. In the following screenshot the Russian language is chosen to input a global search.

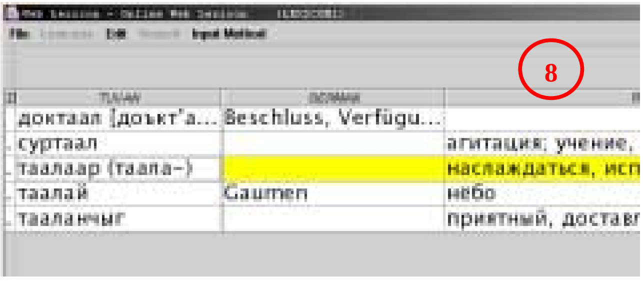




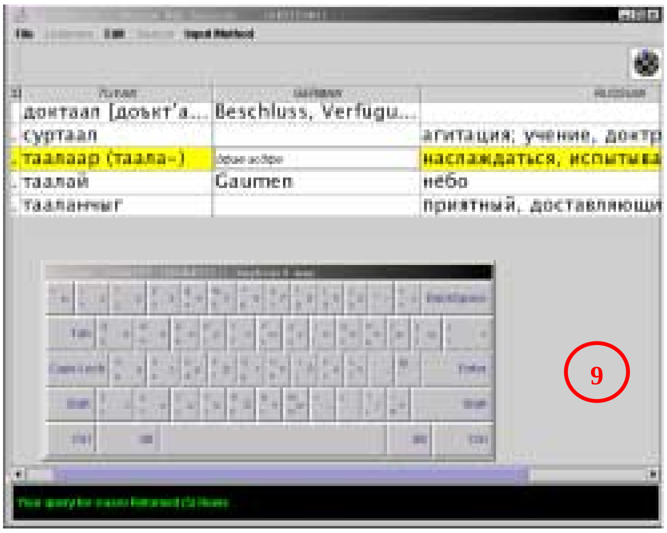
The authorized user may want to decide to modify the dictionary's content. In that case the edit menu offers a number of possibilities.



The user then can select a whole row that he wants to work on.



By double clicking into a cell, he can enter a new term or change an existing one. Again he can choose a certain input method to carry out his operations.



In the similar way, the user could create a new entry.

5. Experiences

Researchers carried out the first tests. The bootstrapping of the online lexicon was done by converting Excel spreadsheets into database tables. Of special concern was whether the Cyrillic characters were converted correctly. It turned out that the conversion was correct, but not all characters that were needed were presented by the input method. The developers tested the multi-user capability, but extensive tests with a larger group of users still need to be conducted.

The lexicon tool seems to be easy enough but can be made more intuitive. For example when a user selects a lexicon no data is displayed which can confuse him. A tabbed interface with easy selections (in alphabetic languages the letters a to z) could improve usability.

For the general user who simply wants to read the lexicons, it was deemed important that a simple interface is available from a HTML based interface. This would not require the user to first download and install an extra tool. Although the JNLP framework simplifies loading and installation, for many that sort of operation forms an obstacle.

People from the indigenous communities did not yet test the tool. So nothing can be said about the usability of the user interface for such a group of people.

The tool is ready be used. Further trials will determine the functionality that should be implemented.

6. Future Perspectives

A number of improvements are obvious for the next version. The next step is as mentioned above, is the addition of a simple HTML interface to allow passive operations on the lexicon. Another wish that can be accommodated easily is to provide more robust and user configurable input methods.

Further, the components have to be chosen such that a local operation without Internet connection is possible. This is necessary to support field researchers during their field trips. Remote operation, however, creates another dimension for the version management. A lexicon update becomes a matter of merging between two lexicon versions that may have developed separately for a few months. Such a merging can only be solved by organizational means. The availability of a difference tool seems to be necessary to help the primary editor during merging. The merging must be based on an audit trail and rollback functionality so that the primary editor can always decide to go back to a certain version.

The researchers may want to include comments or ratings to flag entries, for example, in cases where the researcher is unclear about his coding or where the translation is correct. This means that the researcher needs the ability to create comment columns. The access permissions should be set per column to allow others who are not members of the team to contribute as well.

Due to the researchers request, currently only a Spreadsheet format is supported as an input format. Lexicons for endangered languages are mostly developed in Shoebox, Word documents, or in some database program. Thus, the user should have input possibilities for such formats as well. However, lexicons designed with such tools in general have more complex structures than just two-dimensional spreadsheets. For 2-dimensional spreadsheets it is trivial to automatically generate a simple visualization layout. However, to generate an understandable user interface for databases that have complex structures and that can change dependent on the user needs is a difficult task. This intended extension has aspects that are not solved yet.

At the system architecture side we want to replace the limited CGI mechanism with Java Server Pages. This offers much better performance in multi-user environments. Another useful extension is to extend the protocol mechanism between client and server to SOAP that would offer a standardized API that can be used by other lexicon services.

Acknowledgement

The development of the program and the creation of this paper was done in collaboration with the ASLEP team within DOBES, in particular with David Harrison.

7. References

[1] DOBES: http://www.mpi.nl/DOBES

[2] P. Wittenburg, W. Peters, S. Drude (2002) Analysis of Lexiconsl Structures from Field Linguistics and Language Engineering. In Proceedings of the LREC 2002 Conference. Las Palmas, Spain.

[3] C. Manning: KirrKirr Lexicon: http://www.sultry.arts.usyd.edu.au/kirrkirr

[4] GUK: http://gate.ac.uk

[5] JNLP: http://java.sun.com/products/javawebstart/download-spec.html