Electronic Dictionaries – from Publisher Data to a Distribution Server: the DicoPro, DicoEast and RERO Projects

Andrei Popescu-Belis, Susan Armstrong, Gilbert Robert

ISSCO / TIM /ETI, University of Geneva
40, bd du Pont d'Arve
CH-1211 Geneva 4, Switzerland
{andrei.popescu-belis, susan.armstrong, gilbert.robert}@issco.unige.ch
http://www.issco.unige.ch/

Abstract

This article describes a set of initiatives in the domain if electronic dictionary distribution. Their basis is the DicoPro server, which enables secure access to dictionary data on a server. In the DicoEast and RERO projects, the goal is to acquire high-quality publisher data, convert it into numeric format, and provide access to dictionary entries for the participating institutions. We analyze the various problems that appear throughout this process and describe the solutions we found.

1. Introduction: Dictionary Distribution on Computer Networks

The use of computer interfaces for dictionary consultation has increased the efficiency of lexical searches both in bilingual and monolingual dictionaries. Several factors however hinder users from taking full advantage of such tools: incompatible formats between CD-ROM interfaces, need to duplicate data, risk of losses on publisher side due to illegal copies, etc.

Dictionary consultation through a network answers these problems, provided a universal interface is designed, security issues are addressed, and publisher data is effectively acquired and converted to machine-readable format. These were the main goals of a set of projects initiated by ISSCO. In this paper, we will focus on more recent developments, starting first with a description of these projects, namely XDico/DicoPro, DicoEast and RERO (Section 2). We then study the data conversion process, from publisher data to the numeric format used in the DicoPro server (Section 3): the conversion relies mainly on XML stylesheets (XSL). Next, the dictionary server is described from a technical point of view (Section 4). And finally, we outline some research perspectives opened up by this work (Section 5).

2. Challenges from Three Operational Projects

2.1. XDico and DicoPro

The origins of the DicoPro server (Armstrong et al. 2000) go back to the 1980's (Robert & Petitpierre 1997), but the present version of the server was developed in 1998-1999 as a project within the European MLIS Program. It was developed at the University of Geneva. The data was initially acquired in the 80's for research purposes in lexicon development and NLP and required conversion of type-setting data. Once converted, it became clear that this data would also be useful to the University community at large. Note that at this time Internet services were not widespread and on-line resources a rarity.

The first version of the 'Dico' architecture is still in use on our university's intranet. The long experience with this application shows that publishers are willing to provide data for `internal' use if security concerns are addressed as they were in this project.

2.2. The RERO Project

Based on the success with the DicoPro server at the University, we were contacted by the library service association, RERO (Network of Libraries from Western Switzerland), to install DicoPro for the community they serve. This community includes all higher institutions of learning in Western Switzerland with a potential user base of some 40,000 students and staff.

New dictionary data was added, in particular, a large English dictionary for non-native speakers as well as medium-sized bilingual dictionaries. This entailed new negotiations with the data provider (HarperCollins publishers) for a general license agreement for the academic community served by RERO. The server was modified according to specifications provided by RERO. The data is encrypted and secure on a RERO server and access is controlled by IP addresses. Cf. sections 3 and 4 for a more detailed description of the data and the server.

The server has been up since the beginning of the year and initial reactions have been positive. A log file of user access and queries is kept that should serve to give insight on how such a service is used. In the future we hope to add new data, but have found that publishers are still quite wary about making data available over the Internet via a third party.

2.3. The DicoEast Project

The DicoEast project, supported by the Swiss National Science Fund under the SCOPES program, aims at modernizing the lexicographic infrastructure and methods by building a dictionary server that could be used in the three different institutions involved in this project. The dictionary tool is the DicoPro server provided by ISSCO, University of Geneva, and the Eastern European partners are the RACAI and LMD institutes from, respectively, the Bulgarian and Romanian Academies. The dictionary data consists of freely available dictionaries for Western languages, and from Bulgarian and Romanian dictionaries purchased within the project.

More specifically, the following developments were proposed for this three year project (2001-2003):

- Acquiring the Bulgarian and Romanian electronic form of the printed dictionaries from the copyright holders and converting them into HTML format for use with the DicoPro tool. Negotiation of copyrights for dictionaries that are in unstructured electronic formats.
- Processing of special characters (Romanian) and of whole non-Latin alphabets (Cyrillic). Portability of the server to recent OSs with multilingual support. Installation of the DicoPro tool and the new dictionaries in the two partner institutions.
- Conversion and installation of the new dictionaries for access inside the participant laboratories.
- Improvement of the curricula for the Master degree program by including specifics of DicoPro methodology and training.

At the University of Geneva, the Romanian and the Bulgarian resources will thus be made available. This will contribute to the diffusion of basic resources for non Western European languages both for research and education. The project will also contribute to extending the dictionary consultation tool to a larger audience with more data, in more varied formats, character sets and languages.

The achievements of the first year of the DicoEast project lie mainly in the establishment of a common dictionary framework, both at the level of dictionary data and dictionary software. For the Eastern European partners, the DicoPro software that was installed and explained thanks to the DicoEast project provides the basis for dictionary distribution. A common know-how for dictionary formatting has been shared, as well as the principles of copyrighting and distribution, synthesised from the Swiss partner's long-standing experience. Beneficial relations have been established with other international projects in favour of Eastern European languages. The project working meetings and training meetings have provided the partners with the opportunity to enhance their competencies in dictionary servers and computational lexicography.

3. Conversion of Publisher Data to XML and HTML

The DicoPro server answers queries for dictionary entries by providing HTML-formatted dictionary data for display. The data is stored in encrypted form on the server, but it must be provided to the installers as one HTML file per dictionary. In reality, the file contains a series of <ENTRY> elements, each one having as an attribute the headword of the corresponding entry. The main conversion task must therefore transform the publisher's data into HTML that is well formed and has a user-friendly aspect for display.

Two main difficulties appear, in general, in the conversion process from various in-house formats to a machine readable standardized format: a conceptual and a practical one (see also (Dillinger 2001)). Since publishers often use in-house formatting guidelines, their tagging conventions must be converted to more transparent formatting, easily convertible to HTML. But the data itself does not always respect the in-house guidelines—especially for entries with complex lexical phenomena—therefore it must undergo some manual editing. This verification is also beneficial to the publisher.

3.1. The Conversion Problem

The conversion problem which occupies us here is easy to describe. In all of the three projects, the publishers provided a set of files containing the data, generally in annotated text format. We will focus here on the formatting developed for the RERO project, described at length in (Popescu-Belis, 2002). The publisher (HarperCollins Ltd.) provided six bilingual dictionaries (all combinations of English, French, and German). Each of them has about 25,000 headword entries, which are published as middle-sized paper dictionaries. The initial data was made up of six sets of 26 files each, one per letter, all of them in *text format*, accompanied by separate guidelines describing the publisher's annotations.

3.1.1. Initial Format of the Data

The *publisher annotation conventions* that define the format of the initial data play a central part in the conversion process. Each of the entries consists in a series of lines, each line starting with a label or tag, followed by an elementary piece of information, related to the entry, of the type described by the tag. The first line contains the headword, which, depending on its type, may be tagged <HWME>, <HWKE> or <HWAE>. The structure of each entry is thus embodied in the sequence of lines and tags, as illustrated in the following example of the 'ache' entry.

```
<HWME> ache
<PRON> eIk
<POSP> n
<TRAN> mal $
<TGGR> m
<TRAN> douleur $
<TGGR> f
<POSP> vi
<LBSN> be sore
<TRAN> faire mal
<TRAN> être douloureux*
<TRSB> euse
<LBSN> yearn
<HWXT> to ache to do sth
<TRAN> mourir d'envie de faire qch
<PHRS> I've got stomach ache {or} > a
stomach ache
<LBRN> US
<TRAN> j'ai mal à l'estomac
<PHRS> my head aches
<TRAN> j'ai mal à la tête
<PHRS> I'm aching all over
<TRAN> j'ai mal partout
```

Figure 1. Source text for 'ache' entry

3.1.2. Publisher's Annotation Guidelines

The *guidelines* (written explanations) to the annotation conventions were provided to us by the publisher. They are of course essential in order to understand the annotation mechanism and to write appropriate initial conversion scripts. It is quite strange to note, on one hand, the high degree of accuracy (i.e., of conformance to the guidelines) of most of the 172,012 entries, and on the other hand the fact that there were still several hundred

entries that have smaller or bigger anomalies—that is, whose structure does not match the one dictated by the guidelines.

Below is a short excerpt of the guidelines for the <LBIN> tag. It is quite clear that these guidelines strongly rely on the sequential order of the tags. The use of special marks ('$', '>' and '*') to indicate the place where the content of some tags must be inserted into others (as in Figure 1, line 16) further complicates the formatting task.

```
<LBIN> - meaning label, general. To be
     output in italic within round roman
     brackets, preceded and followed by a
     character space, unless following a
     <TR..> or <XR..> tag where it would be
     preceded by a semi-colon and character
     space.
```

Figure 2. Annotation guidelines for <LBIN> tag

While most of the tags are identical in all of the six dictionaries, there are some differences between them that are not only due to linguistic matters, but sometimes only to arbitrarily different conventions. Tags that are particular to each dictionary do not pose such a big problem as tags that have the same name but different semantics in different dictionaries. A separate XSL stylesheet had to be written in this case. The encoding of the phonetic characters and of the iso-latin-1 character set will not be discussed here—see (Popescu-Belis, 2002).

3.2. The Conversion Process

3.2.1. Overview

The numerous stages of the conversion process have been grouped into several scripts, that are triggered hierarchically. The main stages are:

- Preprocessing of the publisher data to convert it to well-formed (but "flat" or unstructured) XML files. It is during this stage that the source files had to be edited, so that all incoherent data is brought to a coherent, processable form. These changes have been logged and sent back to the publisher, being useful to them for improving the quality and coherence of their data.
- Conversion of the flat XML files into better structured, displayable HTML files. The publisher tags are converted here to HTML tags using the XSL (XML stylesheet language) mechanism.
- Final processing of the special characters, so that they are accepted by the DicoPro server, and concatenation of the a-through-z files for each pair of languages, so that the files are ready to be fed into the server.

3.2.2. First Step: Conversion to XML

Our first goal was to convert the publisher data to well-formed XML, using minimal processing, i.e. the implicit structure of the data was not made explicit at this stage, but was left embodied in a flat XML structure. However, the '*', '$', and '>' codes were processed at this stage. Figure 3 shows the 'ache' entry under XML format.

```
<ENTRY>
<HWME>ache</HWME>
<PRON>e&#x26A;k</PRON>
<POSP>n</POSP>
<TRAN>mal <TGGR>m</TGGR></TRAN>
<TRAN>douleur <TGGR>f</TGGR></TRAN>
<POSP>vi</POSP>
<LBSN>be sore</LBSN>
<TRAN>faire mal</TRAN>
<TRAN>&#234;tre douloureux<TRSB>euse</TRSB>
</TRAN>
<LBSN>yearn</LBSN>
<HWXT>to ache to do sth</HWXT>
<TRAN>mourir d'envie de faire qch</TRAN>
<PHRS>I've got stomach ache <i>or</i>
      <LBRN>US</LBRN> a stomach ache</PHRS>
<TRAN>j'ai mal &#224; l'estomac</TRAN>
<PHRS>my head aches</PHRS>
<TRAN>j'ai mal &#224; la t&#234;te</TRAN>
<PHRS>I'm aching all over</PHRS>
<TRAN>j'ai mal partout</TRAN>
</ENTRY>
```

Figure 3. XML formatting of the 'ache' entry

3.2.3. Second Step: Conversion to HTML

The final result is HTML code – only the <BODY> element more exactly – that is to be displayed using the interface window in a web browser. The criteria defining the aspects of this formatting are: resemblance with the paper version, adaptation to the computer display (greater space is available than on the compact paper form, colors are available too, etc.), intrinsic coherence, readability, clarity, etc. The final aspect as seen by the end-user is shown in Figure 4.

ache [eɪk]

… ***n***
→ mal *m*
→ douleur *f*

… ***vi***
(= *be sore*)
→ faire mal
→ être douloureux(euse)
(= *yearn*)
to ache to do sth
→ mourir d'envie de faire qch
I've got stomach ache or (US) a stomach ache
→ j'ai mal à l'estomac
my head aches
→ j'ai mal à la tête
I'm aching all over
→ j'ai mal partout

Figure 4. Resulting display of the 'ache' entry (HTML)

To obtain this kind of formatting, one must convert the XML tags to HTML tags that, interpreted by the web browser, will lead to the formatting shown above. This process relies heavily on the XML stylesheet mechanism (XSL) described below. The extension of this mechanism in order to produce a more structured XML output, that

could prove useful to computer applications, is discussed in our last section.

3.2.4. The Use of XML and XSL

Once converted to well-formed XML format, the dictionary data is ready for further processing by tools related to the XML standard. We do not check the XML files for validity, since we do not attempt to write a DTD for this format. As the <ENTITY> elements are completely flat, the validity test would be of very little use. At present, the main task is to convert these XML files to HTML files displaying the dictionary data in a coherent and user-friendly format.

The XML-Trans formatting tool developed at ISSCO (Walker et al., 2000) was initially used for this kind of data conversion. Despite its intrinsic qualities, we preferred for this new series of dictionaries to use the XSLT standard (eXtensible Stylesheet Language Transformations), which has now reached maturity and considerable expressive power. XML-Trans was based on transformation rules expressed using regular expressions, simple operations (replacement, insertion, etc.), and pattern-matching, whereas XSLT is more declarative and more XML-oriented, in that it processes XML elements, and allows functions and conditional branches to be written. The operations performed by the XSL files were outlined above. Script-based pre-processing was still needed in order to process some insertion marks; in this way, the code was much shorter to write.

3.2.5. Results of the Conversion Process

At present, the conversion of the six bilingual dictionaries has been completed. A significant number of corrections or changes had to be made in the source data for the conversion process to go on automatically without unpredictable breaks, and to produce coherent data. So, using the modified sources, the conversion process—scripts and XSL transformations—performs automatically on all the six dictionaries. The conversion of the entire set of 172,012 entries takes about ½ hour on a SunBlade100™ workstation (Solaris 2.8).

The final HTML formatting could of course be improved. Some may feel that our choices for the display are not as clear as possible, and small incoherencies (such as bold for italics, or missing commas) have been observed in a very small number of entries (estimated at less than 0.1% of the total). Given the important programming effort that would be needed to process these changes automatically, it is probably easier to only spot them automatically, then change them manually in the HTML files.

4. The DicoPro Secure Client/Server Architecture for Dictionary Distribution

The DicoPro architecture has now reached a stable state that enables ISSCO to use it in application projects. We describe here the main features of the server application, of the client interface and of the security elements (data licensing, encryption and consultation (Petitpierre & Murphy 1997)). The free distribution of the program with an open-source license is currently under consideration.

There is certainly interest and possibilities to freely distribute the source code. However, the main issue is the cryptography module, linked to security considerations. Two possibilities are under consideration, distribution without it (but still with full access control) or development of a new module, possibly within the open-source community.

4.1. Behind the scenes

4.1.1. Data

In order for the indexer to extract individual entries from the HTML output, comments need to be added to the HTML code.

```
<!-- BEGIN -->
<!-- ENT=headword -->
Text of the entry
<!-- END -->
```

The main indexes are automatically created when the entries are processed. The license file for the dictionary must be prepared in order to encrypt entries. The HTML output file and its associated file of headwords and offsets are first fed to a local SQL server and then fed back to files in a database export format. The SQL data files and all related files are then packaged in a single archive file for different platforms.

4.1.2. Server/client installation

The DicoPro server is based on servlet technology -- a standard technology in Web development. The server is written in Java, a very popular, platform-independent, standard object oriented language. The server package is installed and configured in an Apache Web server using the SSL encryption module.

We use a MySQL server as a backend to store the data. The server sends a request to the SQL server, which sends the solution back to the client through the decryption module. The choice of these standard technologies allows for a system independent and portable architecture. Thus the server can be installed on a variety of platforms such as MSWindows, Linux or Unix.

The client side is an applet also written in Java. This applet is sent from the Web server to the browser when the user is connected to a DicoPro Web page. This only requires that Java 1.3 plug-in is installed on the computer.

4.1.3. Security matters and configuration files

Security was a very delicate issue in the DicoPro MLIS project. In order to entice editors to distribute dictionary data via such a server, security must be guaranteed. Security issues include control of user access to the server and the protection and encryption of the data.

Access controls are parametrizable on the server side. The Apache Web server technology offers refined control mechanisms such as limiting access to authorized IP clients. The DicoPro server can also limit the number of simultaneous connections or the number of entry downloads. These mechanisms offer flexibility for implementing different data licensing schemes.

The encryption of the data is enforced by the definition of a unique licence key. This licence key is used as the starting point for the encryption of the data in the SQL database. Among the numerous licensing schemes available, we opted for this solution as the easiest to

implement for the given situation. This solution of a site license reflects a payment scheme where the price is proportional to the number of potential users of the service. It is of course possible to install publicly available data in an open manner without any encryption.

Once the license file for the dictionary has been prepared (see Preparing the Dictionary License) the signature and secret dictionary key must be extracted. The license signature is used to verify that the license file has not been tampered with. This ensures that a dictionary cannot be used outside of the license parameters. The secret key which is used to encrypt the dictionary is generated as a function of this signature. Thus, even if a user is able to tamper with the license file and generate a valid signature, he or she will still not be able to decrypt the dictionary data.

4.2. Connection and Use of the Server

4.2.1. Step one: the DIS server

The Dictionary Information Service (DIS) is a portal to a series of dictionaries. The DIS allows a user to browse several dictionaries, possibly coming from heterogeneous sources. The DIS also displays publisher information about the data, a strong commercial argument.

4.2.2. Step two: dictionary consultation

Options for dictionary consultation are based on a powerful pattern-matching mechanism including prefix matching, exact matching and regular expressions. The indexing constraints on the database are currently indexed on headwords only though other fields could be added depending on the nature of the mark-up. By default, dictionary lookup is set for "Prefix" search mode, a method commonly employed for electronic dictionaries.

Another possibility is a search mode by patterns. In this case, the search key describes the primary keys of the entries to select by means of a regular expression pattern. This can be useful in case of uncertainty in spelling a word, searching for occurrences of words with a given suffix or as an aid for crossword puzzles.

Another search mode is case and accent insensitive query. As it is often cumbersome to type letters with diacritic marks (such as accents), the possibility is given to specify search keys where a plain letter stands for itself as well as any of the associated accented characters. For example the search key "eleve" will match the words élevé as well as élève. This works by mapping the accented characters of both the search key the primary keys of the lexical entries to the corresponding non accented letters.

5. Perspectives

5.1. Future Work

One of our main perspectives is the conversion of the present data to a more structured format, which better reflects the content of a lexical entry rather than the form to be displayed. This format could increase productivity on the publisher side, as well as boost applications thanks to high-quality data. Indeed, such resources have numerous applications in computational lexicography, and information technology in general; computer-aided translation seems one of the most promising ones. The resource could be distributed or sold through an appropriate organization (such as ELRA or LDC) to companies wishing to use it commercially, or to research institutes, while the publishers still retain their copyright. As for in-house uses, such resources would simplify the dictionary revision and updating process, while being easy to convert to export/display formats such as HTML, for use on servers or on CD-ROMs.

The present dictionary data could be converted to a more conceptual form using the above conversion protocol. This form would reflect in a more accurate way the *conceptual structure* (the *content*) of a lexical entry. The XML elements would be embedded according to their logical dependencies, which would significantly depart from the present flat structure, in which dependencies are expressed procedurally, using the rules given in the formatting guidelines. Such a *conceptual formatting* would require the definition of a variety of XML tags, including also attributes that could store part of the information regarding elements. A DTD or XML-Schema would be required to check the validity of the result.

It is to the XSL stylesheet that the most important effort should be dedicated. Given the present state of the stylesheet, roughly half of the work is already done, that is, the interpretation of the sequential order of the source (Collins) markup. What needs to be differentiated, as future work, is the set of replacement tags, as well as the structure of each element in part, since the output options would no longer be the five or six tags available in HTML, but a much greater set of tags.

5.2. Starting Points for Future Work

Several proposals already exist in this direction. One of the best knows formats is the TEI directives with the DTD for (printed) dictionary encoding, a valuable starting point for future work (Sperberg-McQueen and Burnard, 1999: chapter 12). More recent initiatives such as the CONCEDE project have developed and criticized the TEI format for dictionaries.

The OLIF Consortium has been formed to develop the Open Lexicon Interchange Format (OLIF, 2001), which is a user-friendly vehicle for exchanging terminological and lexical data. This encoding format is XML-compliant, and covers a wide and detailed range of linguistic features related to lexical entries and terms. It thus offers support to various natural language processing tasks (such as machine translation).

The Computational Lexicon working group of the ISLE Project has defined the format of a multilingual lexical entry, to be used in computational applications (ISLE CLWG, 2001). This work tries to extend existing work on computational monolingual lexica, such as WordNet and EuroWordNet, to multilingual lexica. The entries defined in this working group can be encoded in XML format, and a tool has also been developed for database-to-XML conversion.

The success of cooperative projects for open-source software (such as Linux) indicates that a similar, collective, approach could work for the development of lexical resources. A proposal by (Mangeot-Lerebours, 2001) describes a framework that handles and displays large 'lexical databases', allowing cooperative development. The process leading from various lexical

resources, available under electronic form, towards the development of machine-readable bilingual dictionaries has also been analyzed by (Dillinger 2001). This work highlights the importance of high quality input data, as well as the necessity of automatically determining the entries that need hand-vetting (manual revision). This procedure parallels the one adopted in our own task.

Therefore, it seems indeed to us highly advisable to extend our current work by converting the publisher's data to a machine-readable format which encodes relations in a structural, not procedural way. The main strength of the bilingual computational resources we processed is the high quality of the input data, which exceeds that of most existing computational resources, since it is based on the work of a team of professional lexicographers.

6. Conclusion

The present work has many implications for computational lexicography and natural language processing (NLP). It appears that high-quality dictionary data, as provided by the publishers through their long-term lexicographic effort, does not come in directly usable machine readable format, but has to be converted from a "paper-oriented" format. Still, this data would be extremely valuable to NLP applications. For instance, machine translation would benefit from high-quality machine-readable bilingual lexicons. Our projects have shown that formatting this kind of data is tractable but needs an initial effort. However, publishers are reluctant to give access to this data unless security concerns are properly addressed.

7. References

Armstrong, S., C. Brace, D. Petitpierre, G. Robert and D. Walker, 2000. DicoPro: An Online Dictionary Consultation Tool for Language Professionals. *Proceedings of Euralex 2000*.

Dillinger, M., 2001. Dictionary Development Workflow for MT: Design and Management. *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain, pp. 83-88.

ISLE Computational Lexicon Working Group, 2001. *The Multilingual ISLE Lexical Entry (MILE): a Discussion Paper*. http://www.ilc.pi.cnr.it/EAGLES96/isle/cldoc/02_ISLE_WP2-WP3_DISCUSSIONPAPER.ZIP.

Mangeot-Lerebours, M.,2001. *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue.* Ph.D. thesis, Université Joseph Fourier, Grenoble.

OLIF Consortium, 2001. *Proposal for the Structure and Content of the Body of an OLIF2 File*. http://www.olif.net/olif2/documents/specificationJuly2001.pdf.

Petitpierre, D. and D. Murphy, 1997. *Proposals and Specifications for Licensing Schemes.* DicoPro Project Report ISSCO, Geneva, D4.2.

Popescu-Belis, A., 2002. Conversion of Bilingual Dictionaries to HTML Using XSL. Technical Report 3 (2002), ISSCO/TIM/ETI, University of Geneva 28 p. (public version).

Robert, G. and D. Petitpierre, 1997. Dico: un outil de consultation de dictionnaire en réseau. *META*, vol. XLII, n. 2, pp. 283-290.

Sperberg-McQueen, C.M. and L. Burnard, 1999. *Guidelines for Electronic Text Encoding and Interchange*. TEI P3 Text Encoding Initiative, Oxford.

Walker, D., D. Petitpierre and S. Armstrong, 2000. XML-Trans: a Java-based XML Transformation Language for Structured Data. *Proceedings of COLING 2000*, Saarbrucken, Germany.