DIFFERENT ISSUES IN THE DESIGN AND DEVELOPMENT OF THE ELECTRONIC CUBAN BASIC SCHOOL DICTIONARY

IÑAKI ALEGRIA*, XABIER ARREGI*, XABIER ARTOLA*, MIKEL ASTIZ*, LEONEL RUIZ MIYARES**
University of the Basque Country, San Sebastian, Basque Country*
Centre for Applied Linguistics, Santiago de Cuba, Cuba**
xabier.artola@ehu.es

1 Introduction

The Centre for Applied Linguistics (http://www.santiago.cu/hosting/linguistica/) is an Institution of the Ministry of Science, Technology and Environment in Santiago de Cuba and has been working in Lexicology and Lexicography fields since 1973. The centre has finished a large amount of work in lexicology and lexicography over the years (Miyares, 1996, 1998; Marconi *et al.*, 1999). Among others, it is worth mentioning the *Diccionario Escolar Ilustrado* (Miyares, 1998), which is well-known and used in Cuban primary schools.

The *Diccionario Básico Escolar* (DBE) (Miyares, 2003) is the second pedagogical dictionary in Cuba for young students. It is made on basis of lexicology research on a corpus of 700,000 words taken from 7,000 Cuban student compositions, a selection of Cuban popular books and children's literature, as well as relevant Cuban newspapers such as *Granma*, *Juventud Rebelde* and *Trabajadores*. The dictionary contains more than 7,000 entries and 14,000 meanings spread over 1,016 pages in its paper version, and has been awarded with the Laurence Urdang International Award from EURALEX for the support of lexicographical research in 2002.

The electronic DBE project is the result of a collaboration of the Centre for Applied Linguistics (CLA) and the IXA NLP Research Group of the Faculty of Computer Science of the University of The Basque Country in Saint-Sebastian (http://ixa.si.ehu.es)[1]. The investigation area of the group is natural language processing, going from the lexicon and morphology to syntax and semantics.

The main goals of the project included:

- To devise and to apply a method to semi-automatically convert a conventional dictionary into an XML-based dictionary database, focusing on an efficient and structured way of storing and retrieving the information.

[1] The project has been possible thanks to the financial support of the Basque Government (FOCAD project no. PRO-2003K2/0007).

- To develop an electronic dictionary application for secondary and high school students, employable both on CD-ROM and through the web. The electronic version of the DBE has a potential user group of 500,000 students in Cuba, being free to use more than 50,000 computers.
- Finally, to design and develop a dictionary editing environment which will enable a lexicographer to maintain and update the DBE in an easy and flexible manner.

This article explains the conversion process of the dictionary from its original format into an XML-encoded version, along with the design and the implementation details of an electronic dictionary application running on this encoding. After this introduction, the original dictionary and its main features are described in section 2. In section 3, the conversion process carried out to take the dictionary from its original RTF[2] format to TEI-conformant XML is explained. Next, section 4 is devoted to describe the architecture and GUI of the application, along with some implementation issues. Finally, future work is depicted in section 5, and some conclusions are given in section 6.

2 Structure and features of the dictionary

The DBE is a Spanish dictionary that is intended for use by students in secondary level high schools in the age of 11 to 17. One important feature that distinguishes it from ordinary Spanish dictionaries is that it describes Spanish as it is employed in Cuba; so, besides "common" Spanish entries it also contains many specific Cuban Spanish entries.

The following fields can be distinguished in the dictionary entries: headword, typical spelling errors (realized as red letters in the headword), pronunciation (for English words used in Cuban Spanish), part-of-speech, geographic, domain and usage style labels, verbs' inflection model, syllabification of the headword, inflected form(s) of the headword (with number or grammatical inflection type, and part-of-speech labels), one or more senses containing definition text, example sentences, often with labels referring to the part-of-speech used in the sentence or other usage notes, and with references to the headword emphasized in bold and underlined meta-linguistic references, synonyms, antonyms, and similar words, usage notes and labels, attached to the definition or to particular examples, and, finally, related subentries, such as locutions and other noun or verb phrases.

The DBE contains 7473 entries including 14013 word senses. Attached to the entries you can find 1380 diminutives, along with plural forms of nouns, feminine and plurals of adjectives, participles of verbs, etc. Regarding lexical relationships,

[2] Rich Text Format.

3601 synonyms, 474 antonyms and 75 similar words can be found in the dictionary, allowing to navigate between related entries. Moreover, 1062 locutions, 651 phraseologisms and 39 sayings are defined and exemplified as subentries of the main entries.

The dictionary was originally typed at the CLA and stored in 27 files in RTF, one per letter. The following sample entries might make clearer the usage of the different fields and features:

cerca sf. Valla, tapia o muro generalmente de alambre, estacas o piedras que se pone alrededor de cualquier terreno para resguardarlo o limitarlo. *Pusieron una **cerca** alrededor del terreno deportivo para evitar que penetren intrusos.*
cer-ca; cercas (pl.); cerquita (dim.)

cerca adv. l. Próximamente, inmediatamente, a corta distancia. *Mi casa se encuentra **cerca** de la escuela. Vamos al cine a pie, pues queda **cerca**.* Ant. lejos. // loc. adv. **cerca de**. Aproximadamente, más o menos, casi. *Mi abuela está saludable, aunque tiene **cerca de** noventa años.*
cer-ca; cerquita (dim.)

fábrica sf. **1.** Establecimiento o edificio donde se fabrica algo, en el que existen equipos, máquinas, herramientas, etc., necesarios para producir determinado tipo de objetos. *Ibia comenzó a trabajar en una **fábrica** de zapatos*. Sin. industria, factoría, empresa. **2.** fig. Acción de construir o producir algo. *La colmena es la **fábrica** donde se elabora la miel*.
fá-bri-ca; fábricas (pl.)

3 Conversion process: from RTF to XML

As mentioned above, the first goal of the project was to devise a method to semi-automatically convert the RTF dictionary into an XML-based dictionary database. Both from a typographically and structurally point of view a dictionary is extremely complex. If we want to identify explicitly all complex features of a dictionary for the purpose of electronic use, a good method is the use of a standard mark-up language (Ide *et al.*, 1993), which will then facilitate the use of the dictionary by an application or program. The use of XML to represent the dictionary knowledge in a structured way allowing to explicitly mark-up the different fields in the entries means a radical change with respect to the original RTF version, offering both the lexicographers and the users more and richer possibilities to later on search the information in the dictionary.

In this section, after discussing how we defined a suitable encoding for the dictionary, we will mention the different phases and tools used in the conversion process.

3.1 Establishing the XML encoding to use: the TEI guidelines

The first step, before the conversion, is to define the XML language to encode the dictionary, i.e. the data structure into which the dictionary must be transformed from its original format.

The TEI guidelines (2001) include a whole chapter on how to encode printed dictionaries. We examined carefully the entries in the DBE and observed that almost all of the features present in them were already foreseen by the designers of the TEI guidelines, with few exceptions. Hence, we decided to encode the dictionary following the guidelines, so adopting a subset of the TEI DTD for dictionaries and adding to it some elements and attributes to deal with the unforeseen features. One of the aspects we found was missing in the guidelines was the explicit encoding of typical spelling errors, which in the DBE is realized by means of red letters in the headword, resulting encoded in our enhanced DTD by means of an element named `posErrors` (see example below).

In order to establish more precisely the encoding, we chose a sample of entries representative of the diverse complexity of the dictionary data, and encoded them in XML by hand, thus deciding how to apply the TEI guidelines to encode the different features in the sample entries. This sample, along with a normative document that gathers all the encoding decisions we made on how to use and adapt the TEI DTD to the particular case of the DBE, has turned out very useful in the conversion process and in the later post-processing and quality control phases.

Let us show as an example the entry **fábrica** once converted into final XML:

```
<entry>
  <form>
    <orth>fábrica</orth>
    <syll>fá|bri|ca</syll>
    <posErrors>3</posErrors>
  </form>
  <gramGrp>
    <pos>sf.</pos>
  </gramGrp>
  <form type="infl">
    <orth>fábricas</orth>
    <number>(pl.)</number>
  </form>
  <sense n="1">
    <def>Establecimiento o edificio donde se
fabrica algo, en el que existen equipos,
máquinas, herramientas, etc., necesarios para
producir determinado tipo de objetos.
  </def>
```

```
      <eg>
        <q>Ibia comenzó a trabajar en una <oRef/>
de zapatos.</q>
      </eg>
      <xr type="syn">
        <lbl>Sin.</lbl>
        <ref>industria</ref>
        <ref>factoría</ref>
        <ref>empresa</ref>
      </xr>
    </sense>
    ...
</entry>
```

3.2 The conversion phases

The conversion of the dictionary from RTF to XML has consisted of three consecutive phases:

- Pre-processing of the original documents in order to get an unambiguous, "normalized" and consistently *edited RTF version* of the dictionary. *MS Word* macros have been used in this phase.
- Conversion of the edited RTF version into a *preliminary XML version* using a tool named *Ferret* (Patrick *et al.*, 2002), whose goal is to semi-automatically learn the structure of dictionary entries, based mainly on typographical features, and to encode them into (in our case, preliminary) XML.
- Post-processing of the preliminary XML version in order to correct encoding errors, so getting the *final XML version* of the DBE. XSLT scripting has been used in this phase[3].

One important basic principle that has to be kept in mind during the conversion is the fact that the conversion can never be done perfectly. Because of the complexity of a dictionary there always exist small features which can not be automatically converted in a proper way and have to be corrected afterwards. Of course we have strived to minimize the amount of these corrections, because it can be a time-consuming and sometimes annoying activity, but in the end a human quality control phase has been carried out to guarantee that the final encoding of the dictionary is completely correct.

[3] The XSLT and XQuery Processor *Saxon* (http://saxon.sourceforge.net/) has been used in this phase.

3.3 Evaluation of the conversion process and quality control of the DBE

The evaluation has been based on an exhaustive quality control of the entries, which were manually revised at the application prototype, being their encoding manually corrected when necessary. As a first figure, we can say that 984 entries (13%) have been corrected as a result of this quality control.

At a first stage, a selective control of entries was carried out. The entries reviewed in this stage had been reported by the lexicographer as problematic while priming them with *Ferret*, or by the post-processing script programmer who noticed several irregularities in the preliminary encoding that caused the entries could not be transformed automatically in a consistent manner. Diverse reasons can be found here among the ones used to report an entry as problematic: pre-processing errors and/or inconsistencies, spelling and/or typing errors already present in the printed version, character encoding difficulties, differences in the encoding of emphasis in definitions and examples, etc. Around 100 entries were revised and, if needed, corrected in this stage.

Then, every entry in the dictionary was reviewed by a lexicographer, using for that an early prototype of the application, and the problems found in its rendering were reported. The XML encoding of the problematical entries was then manually revised and corrected. The corrections made at this second stage are also very heterogeneous, and range from the correction of simple punctuation and/or spelling errors up to major encoding changes; in a few cases, new lexicographical elements were discovered in the entries, and this obviously led to make changes and/or additions to the subset of the TEI's DTD adopted for the DBE, and to the rendering of these new lexicographical elements in the application interface. The lexicographers profited from the opportunity of this exhaustive reviewing to correct and enrich many entries (adding or enhancing definitions and examples, including new synonyms or related entries, etc.), thus improving the quality of the dictionary. On the whole, around 850 entries were touched up in this stage, which was followed by a second minor revision (only the entries modified were looked again) that affected nearly 250 entries; in this figure we include some changes made automatically in order to normalize hyphens and dashes, for example, which were found to be realized by means of different characters in the printed version.

4 Electronic DBE: architecture, GUI and information flow

The XML dictionary documents constitute the basis of an electronic dictionary application already developed and distributed at 200 Cuban schools. The main goals of the electronic version of the DBE have been (1) to provide the students with a useful and modern dictionary tool for the learning and practice of the language, (2) to build first a CD version of the dictionary, and then to make it

available on-line in the web, and (3) to make use of XML technology as a basis for the storage and exploitation of the dictionary.

The functionality we wanted for the application included:

- First letter and index-based browsing facilities.
- Normal search or lookup, with closest match help.
- Advanced search: filtering of entries based on selected parts-of-speech.
- Hyperlinking facilities: cross-references between related entries, synonyms, antonyms, etc.
- Orthographic help, based on purposely encoded misspelling feasibility.
- Accessibility of verb paradigms and illustrations directly from the entries.
- Some statistics on the contents of the dictionary and help.

The application has been developed as a web application. The user needs just an ordinary web browser where the GUI of the application is shown. For the CD version, the web server is embedded into the application, while for the on-line version we are considering the use of a conventional web server. Anyway, client-side code should remain the same for both versions, while server-side code, although very similar, will have to be rewritten for the on-line version.

In this section, we will explain the design and architecture of the application, describing how the data are represented and structured, and outline the information flow between the dictionary server and the user interface.

As just mentioned, a client-server architecture supports the application. Requests submitted by the user through the browser are processed at the web server, which in turn replies sending the data to the browser. A standalone *Ada Web Server*[4] has been used for the CD version, and it is also being used already in a prototype of the on-line version, running on top of an Apache server that redirects the dictionary requests to the *Ada Web Server*. Most of the time, the server processes the information it has in XML converting it into XHTML before sending it to the client, so ensuring that the content sent will be properly rendered at the browser independently of its XML-processing capabilities.

4.1 Client-server architecture.

The original dictionary was separated into 27 files corresponding to the 27 letters of the Spanish alphabet. After the conversion into XML the dictionary still consists of 27 separate documents, which constitute the basis of the application. Moreover,

[4] Ada Web Server. © 2000-2003 ACT-Europe. Authors: Dmitriy Anisimkov, Pascal Obry.

indexes, verb paradigms, and illustrations can be found as well on the server side of the application.

So, viewing it at a more detailed level, the dictionary server contains:

- Entry files, adequately encrypted[5]: one document per letter (*a_.xml, b_.xml, ... z_.xml*), where each entry is uniquely identified by the `id` attribute of the `entry` element. As an example, let us show you the entry **fábrica**, as stored in the *f_.xml* document at the application (the content of the `entry` element is encrypted):

```
<entry id="f_d0e148">ba9d23c4 ... 57d</entry>
```

- Indexes: one document per letter. The information attached to the entries in these indexes consists of the identifier and the headword, and the different parts-of-speech it belongs to (in any of its senses). The indexes are both accessed in normal and advanced search modes to (1) populate the headword lists at the application GUI, filtering them based on the choice of POS made by the user in the case of advanced search, and (2) to get the identifier of the entry requested given its orthographical form, to subsequently fetch the corresponding entry from the entry documents. They are also used when providing orthographic help to the user (see below, at section 4.3.1). This is a partial view of the *f* letter index:

```
<entryIndex>
   ...
   <entry id="f_d0e148">
      <orth>fábrica</orth>
      <pos>sf.</pos>
   </entry>
   ...
   <entry id="f_d0e1450">
      <orth>falso</orth>
      <pos>adj.</pos>
      <pos>sm.</pos>
   </entry>
   ...
</entryIndex>
```

- Verb paradigms: 80 inflection models, encoded also in XML, and directly accessible from the verb entries by means of a hyperlink set on the element that indicates the conjugation model according to

[5] Encryption is only used in the CD version.

which the verb conjugates. As an example, this is the verb paradigm of model no. 80, corresponding to the defective verb *concernir*:

```
<model id="80">
   <mode id="infinitivo">concernir</mode>
   <mode id="gerundio"
esp="1">concerniendo</mode>
   <mode
id="participio">concerniente</mode>
   <mode id="indicativo">
      <tense id="pres">
         <conj p="3s">concierne</conj>
         <conj p="3p">conciernen</conj>
      </tense>
      ...
   </mode>
   ...
</model>
```

- Illustrations: 685 drawings and pictures in JPEG format can be found at the current version of the application. These images can be accessed by means of a hyperlink set on the `target` attribute of the corresponding `entry`, `sense` or `re` (related entry) element, or by browsing an illustration index created for this purpose.

As usual in web pages, rendering stylesheets *(.xsl* and *.css*) are located on the server, and they are used, in our case, to transform XML content into HTML before sending it to the client. Moreover, XSLT scripts responsible of fetching entries, indexes and suggestions (in the case of misspelled word searching) are also included on the server side, and constitute, along with the client-side JavaScript functions, the heart of the application functionality. On the client side, the application reacts to the user's actions and to the events occurring when using the dictionary, submitting the requests to the server and displaying the information received from it.

4.2 Graphical User Interface

The graphical user interface (GUI) of the application consists of several multiframe HTML pages. As has been said, client-side scripting is used to respond to the events produced on the interface (user clicks, input through the keyboard, etc.).

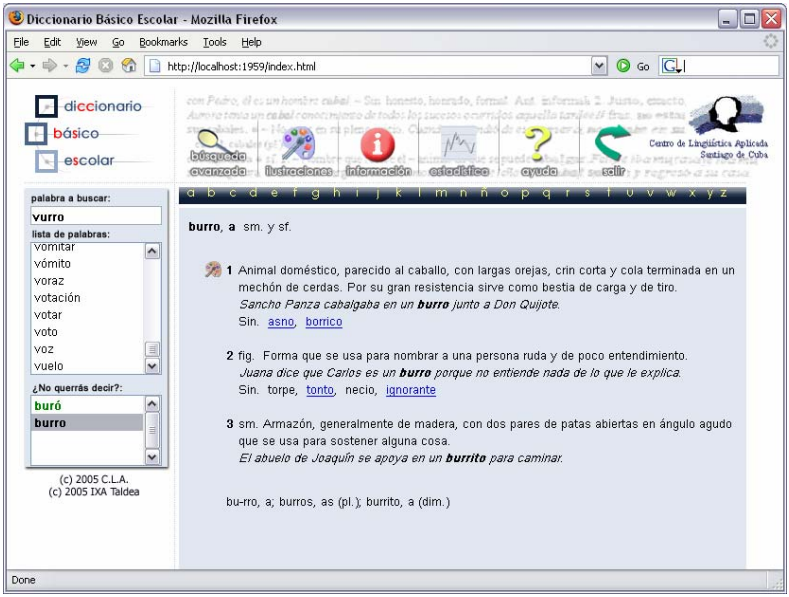

Figure 1. Graphical User Interface of the electronic DBE.

In Figure 1 a screenshot of the main window of the GUI is provided. As can be seen, the GUI contains the following elements: a menu bar that gives access to the different functionalities of the application (statistics, illustrations, help, advanced search, etc.), an alphabet bar for browsing the dictionary based on the first letter, an input textbox or search frame, a headword selection listbox, a suggestion listbox, and a main area for displaying the entries. In some entries, standard-look hyperlinks are displayed allowing the user to navigate from there to other related entries, such as synonyms, antonyms, and so on. Moreover, in advanced search mode a special checkbox bar allows the user to select the parts-of-speech wanted to be used as a filter when filling up the headword selection box.

The interface developed meets both the wishes of the members of the CLA and the requirements of the application, especially regarding the target user group analysis. In the design of the GUI we decided to merge the search and browse functionality. This is because they have a lot of aspects in common and we can combine them without losing the requirement to keep things simple in use. If a user types the first letter of a search string then all the headwords starting with that letter will be shown in the headword selection box. The same thing happens if the user clicks one of the letters of the alphabet bar. Once the headword selection listbox or

index is populated, the user can choose to display one of the entries just by clicking it on the index.

4.3 Information flow: client-server communication.

As mentioned above, the electronic DBE has been developed as a web application, where the server serves XML content in response to requests made by the user through the client GUI. XSLT stylesheets are used on the server to search the information requested and fetch it from the dictionary documents. Moreover, they are also used, in combination with CSS stylesheets, to convert the XML data into HTML before sending them to the browser.

One important remark regarding the underlying infrastructure must be made here. On the one hand, when searching or browsing the dictionary, both in normal and advanced search modes, just the index documents are downloaded in a first moment, in order to populate the headword selection listbox. Filtered index documents are served when in advanced search mode, or when proposing orthographical corrections to the user (in this case to fill up the suggestion listbox). The biggest index file (corresponding to the *c* letter) is less than 100 kilobytes in size, so it is not very costly, in terms of time and space, to completely download one of these each time it is required (with the proviso that, if it is already in the browser's cache, there is no need to download it again); we expect that the behaviour of the application will be similar in the on-line version. On the other hand, dictionary information is served on a per-entry basis, meaning that, when a particular entry is requested, just the XML element encoding it is fetched from the corresponding document, to be then converted into HTML and sent back to the browser to be displayed.

Index documents are fetched from the server, filtered if necessary, and rendered at the browser as HTML `option` elements in the headword selection listbox's select. In the case of entries and verb paradigms, the server must extract the information (the entry or superentry, or the verb paradigm) from the corresponding document (the document matching the initial letter or the document containing all the verb inflection models), convert it into HTML, and send it to the browser. This is performed in both cases in an analogous way: a server function processes the request by invoking an XSLT script, which, based on the parameter(s) provided (the entry identifier or the inflection model number respectively), fetches the required element from the corresponding document, transforms it into HTML by means of a suitable XSL stylesheet, and sends it to be displayed in the main entry display area, in the case of entries, or in a new window, in the case of verb paradigms.

4.3.1 Correction of typical spelling errors

If no exact match is found in the index downloaded (once the user has ended typing), a regular expression representing the possible correct spelling intended by the user is built. This task is based on orthographic criteria (no typing errors are corrected in this version of the dictionary) issued from previous research carried out at the CLA in the area of patterns and likelihood of spelling errors (Miyares, 1996).

Two phases can be distinguished here. The first one, the construction of the regular expression, is performed on the client. The regular expression is then submitted to the server along with the possible starting letters of the correctly spelled candidates and the list of POS selected (in the case of advanced search). The function invoked on the server examines the appropriate index documents, filtered according to the list of POS provided, matching the entry headwords against the regular expression. The server processes the index documents, using for that the SAX model of XML processing, and builds a list of candidates that are then proposed to the user as spelling suggestions.

The construction of the regular expression is based on a set of "clusters" of typical spelling errors. We call cluster in this context to a set of letters and/or pairs of letters that can be misused at a particular position of the word. For example, one of the clusters contains the letters *b* and *v*. In Spanish the *v* is mostly pronounced as a *b*, which explains why this mistake (incorrect substitution of these two letters) is made frequently. These clusters can be used to replace all the letters of the search string that are part of one of them by the other letters in the same cluster, so creating all possible combinations of the search string based on the clusters. The clusters that have been implemented in the current version include the following cases, among others: the above mentioned *b/v* confusion, improper use or omission of *h* before or between vowels, misuse or omission of accent on vowels, incorrect substitution of sibilant consonants (*s*, *c*, *z*, *x*), confusion of *y* and *ll*, misuse of *n* before *p* and *b*, etc. So, when building the regular expressions, every letter or pair of letters in the search string that is member of one of the clusters is replaced by an expression representing the other elements in the same cluster. Obviously, if the letter does not belong to any cluster only the letter itself is inserted in the regular expression. For example, every occurrence of *b* or *v* in the search string will be replaced in the regular expression by *[bv]*, meaning "*b* or *v* can be matched at this position of the word".

In Figure 2, you can see the regular expressions constructed after applying these operations to the search strings *vurro*, *uevo*, and *siudá*, along with the initial letters indicating, in each case, the index files that the server must check to build the list of candidates, and the candidates finally suggested:

Search string	Regular expression	Initial letters	Suggestions
vurro	*[bv]h?[uúüw]s?(l|r|rr)h?[oó]d?*	*bv*	**buró, burro**
uebo	*h?[uúüw]h?[eé]s?[bv]h?[oó]d?*	*huw*	**huevo**
siudá	*[csxz]h?[iíy]h?[uúüw]s?dh?[aá]d?*	*csxz*	**ciudad**

Figure 2. Building regular expressions to correct spelling.

As can be seen, only one regular expression is needed for one search string, and the matching operation of this regular expression against all the headwords in the possible index files is very efficient. Indeed, much more efficient than applying typical spelling correction algorithms, such as the replacement of letters, the transposition of a pair of contiguous letters, and so on. Moreover, using this technique, although not being able to correct typos (it was not our aim to do that, because, in fact, this is not a spelling corrector, but a tool whose goal is to assist the student in the learning of the language), we are able to make "more intelligent" suggestions that would not be possible using those spelling correction techniques. For instance, the spelling corrector included in the Spanish version of MS Word 2003, although it proposes **buró** and **burro** for *vurro*, it is not able to propose the correct candidates for *uebo* and *siudá*, and this kind of errors are of relative high frequency among students.

We think that this approach suits better the didactic character of the electronic DBE, as regards its aim as a tool to help in the learning of the language. Other features already mentioned, such as the use of red letters in the headwords to indicate the likelihood of orthographical errors, were also conceived, already for the printed version, with the same idea in mind.

4.4 Some considerations on the application development and implementation aspects.

It was desirable that both versions, the standalone and the on-line version, shared the same architecture using even the same code as far as possible, so we would only have to implement the application once. Because of the differences in the underlying infrastructure, the two versions would not be exactly equal, but by implementing everything in a modular way, the differences could be kept very small. The only difference between the two versions is that the CD-ROM version works completely on a local machine, and the web version will do a part of the processing on the server machine and send the results to the client computer.

Open-source software has been used to implement the application. On the server side, the main functions are implemented in standard Ada, making use of the Ada

Web Server, XML/Ada[6], Ada_Xslt[7] and Serpent Blockcipher[8] packages, compiled using GNAT, and XSLT scripts are used as it has been already explained above (the libraries *libxml2*, *libxslt* and *libexslt* developed by the GNOME project[9] are used to interpret this scripting). On the client side, JavaScript scripting is all we need to make the application work.

5 Future work

As has been said, short-term future work includes the development and setting up of the on-line version of the dictionary at the CLA website (an already operative prototype is being tested). This will have the great advantage over the CD version that it will be constantly enriched and corrected.

At a longer term, the production of a second version of the electronic DBE is envisaged. This version should include, among other features, full lemmatization of definitions and example texts, in such a way that possibly dynamic hyperlinks over every single word occurring in these texts would be made feasible. A Cuban Spanish lemmatizer is already being developed at the CLA with this goal in mind among others. Such a tool is needed to pre-process the definitions and examples of the dictionary lemmatizing them and making possible the establishment of the links between word occurrences and their corresponding dictionary entries.

Finally, a dictionary editing environment is now a must for the lexicographers at the CLA. We can't anymore edit our old Word files, but we have to deal with XML-encoded documents. We are already working in such an environment, and an operative prototype has already been implemented and is being tested. The requirements of this environment are the following: (1) it must allow adding, deleting and modifying entries in a friendly fashion: XML details must be transparent for the lexicographer; (2) it should provide the lexicographers with all the features of a full-fledged DBMS: full search capabilities, safe storage, concurrent access, integrity constraint checking, etc.; (3) it must allow to

[6] XML/Ada. XML suite for Ada95. © 2001-2002 ACT-Europe.
[7] Ada_Xslt. Ada 95 binding for XSLT. © 2002 Maxim Reznik.
[8] Serpent Blockcipher. Implementation of the AES candidate algorithm Serpent. © 1999 Gisle Sælensminde.
[9] http://www.gnome.org/

- *libxml2:* XML C parser and toolkit developed for the GNOME project. © 1998-2003 Daniel Veillard.
- *libxslt:* XSLT C library developed for the Gnome project. © 2001-2002 Daniel Veillard.
- *libexslt:* Extension library for XSLT. © 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard.

automatically generate the files and components needed by a running application such as the current electronic DBE: entries, indexes, images, etc.; and (4) tailored output must be feasible: it should allow to easy export data required in print editions, diversified electronic versions, etc.

6 Conclusions

In this project, a methodology for the semi-automatic conversion of a dictionary from RTF format into XML has been devised and conducted. The foundation of this methodology is to unambiguously recognize the different fields in the entries, and consists of the application of diverse processing techniques both to RTF and XML. As a result, the DBE is now a real lexicographical database where the information is represented and structured in a suitable way, encoded accordingly to a well-recognized standard such as the TEI guidelines. This database constitutes the basis over which future lexicography work at the CLA will be done.

We must emphasize here the importance of the manual quality control after a semi-automatic conversion process of this kind, because, although costly, we find it essential to ensure the correctness of the data.

Moreover, an electronic dictionary application has been designed and developed. A CD-ROM version has been distributed at the Cuban schools and is already being used, whereas the on-line version will be available pretty soon. We would like to emphasize here two aspects: on the one hand, the architecture of the application that has been developed as a web application, even for the CD version; on the other, its student-orientation, making it a first technology product that can be used at schools as a learning tool, and which follows the research conducted at the CLA over the years on orthography and other language learning aspects (Ruiz & Miyares, 1999).

Another point to highlight in the accomplishment of the project is the collaboration between the two partners, on the one hand the CLA, involved in applied linguistics for more than 30 years, and the IXA NLP Research Group. The long experience in traditional lexicography of the CLA members has met the know-how and experience of previous computational lexicography projects of the IXA group (Arregi *et al.*, 2003), giving as a result this lexicography work and the starting point of a hopefully even more fruitful cooperation in the future.

Bibliography

A. Dictionaries

Marconi L., Miyares Bermúdez E., Ratti D., Rolando C., Ruiz Miyares L. (1999) *Diccionario Ortográfico del Español. Basado en el léxico del escolar cubano.*

Istituto per i Circuiti Elettronici (ICE), Consiglio Nazionale delle Ricerche (CNR), Génova, Italia; Centro de Lingüística Aplicada, Ministerio de Ciencia, Tecnología y Medio Ambiente, Santiago de Cuba (Cuba).
Miyares Bermúdez E. (dir.) (1998) *Diccionario Escolar Ilustrado*. Editorial Oriente, Santiago de Cuba y Ediciones Libertarias Prodhufi, Madrid (España).
Miyares Bermúdez E. (dir.) (2003) *Diccionario Básico Escolar*. Centro de Lingüística Aplicada, Santiago de Cuba (Cuba).

B. Other Literature

Arregi X., Arriola J. M., Artola X., Díaz de Ilarraza A., García E., Lascurain V., Sarasola K., Soroa A., Uria L. (2003) 'Semiautomatic construction of the electronic *Euskal Hiztegia* Basque Dictionary' in Michael Zock and John Carroll (eds.), *Traitement automatique des langues. Les dictionnaires électroniques*, vol. 44-2, pp. 107-124. ATALA (Association pour le Traitement Automatique des Langues), Paris (France).
Ide N., Le Maître J., Véronis J. (1993) 'Outline of a Model for Lexical Databases.' *Information Processing and Management*, vol. 29, no. 2, pp. 159-186.
Meijs K. J. (2002) *Application analysis of the electronic DBE*. Practical Training Report (Faculty of Computer Science, University of Twente). Centro de Lingüística Aplicada, Santiago de Cuba, (Cuba).
Miyares Bermúdez E. (dir.) (1996) *Léxico Activo Funcional del Escolar Cubano*. Centro de Lingüística Aplicada, Santiago de Cuba (Cuba). (unpublished.)
Patrick J., Palko D., Munro R., Zappagina M. (2002) 'User driven example-based training for creating lexical knowledgebases'. *Proceedings of the 2002 Australasian Natural Language Processing Workshop*. Canberra (Australia).
Ruiz Hernández J. V., Miyares Bermúdez E. (1999) *Vacuna Ortográfica VAL-CUBA. Metodología para prevenir y erradicar las faltas de ortografía (Nivel Primario)*. Editorial Academia, La Habana (Cuba), tercera edición.
Text Encoding Initiative Consortium (2001) *The XML version of the guidelines for Electronic Text Encoding and Interchange* (http://www.tei-c.org/).