Semiautomatic conversion of the *Euskal Hiztegia* Basque Dictionary to a queryable electronic form

Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., García E., Lascurain V., Sarasola K., Soroa A., and Uria L.

Department of Computer Languages and Systems
University of the Basque Country
Computer Science Faculty, P.O. Box. 649, E-20080 Donostia
ccpsoeta@si.ehu.es

RÉSUMÉ.

ABSTRACT. *The goal of the project presented here is to implement a prototype of an advanced electronic version of the Basque monolingual dictionary Euskal Hiztegia (EH), hereafter eEH. Here we will focus on two aspects: i) the methodology followed to correct and update the contents of the dictionary after the automatic conversion from the MRD (Machine Readable Dictionary) to the TEI version (Text Encoding Initiative) and ii) the application developed in order to facilitate the navigation and search through the information contained in the dictionary. Our motivation is twofold: 1) to get a well-structured electronic representation of a significant resource for the Basque language to be used as a basic tool in our research and future developments, and 2) to be able to offer common users a set of more useful utilities than those offered by paper dictionaries or classic dictionary browser applications. Here, we include a brief description of the EH dictionary itself, outlining its content and structure. The various stages of the methodology for the eEH are presented by means of examples; on the one hand, we will explain the automatic conversion from the MRD to the TEI version (section 2), and, on the other hand, we will present the steps for the manual correction of the TEI version in section 3. Section 4 dicusses some aspects on the standardisation of the dictionary with respect to the language. In section 5 we will describe the application itself and the types of queries proposed for consulting the dictionary and we will give some examples of use of the prototype in section 6. Finally, some conclusions and future work will be presented.*

MOTS-CLÉS :

KEYWORDS: *Electronic dictionaries, dictionary lookup and indexing, dictionary mark-up, information retrieval*

L'objet – 8/2002. LMO'2002, pages 45 à 57

1. Motivation

The work reported in this article was motivated by two considerations : (1) to adapt a significant lexical resource for Basque representing it in a well-structured way in order to obtain a wide-coverage lexical resource for NLP applications, and (2) the construction of the electronic *Euskal Hiztegia* Basque dictionary (*e*EH) to offer users a set of more helpful utilities than those offered by printed dictionaries.

We considered the *Euskal Hiztegia* (EH) [SAR 96] an adequate source because it is a general purpose monolingual dictionary that covers standard[1] Basque. This dictionary is actually an important reference resource for the study of Basque. It contains 33,111 entries and 41,699 senses, and the examples given within the entries are collected from significant literary texts.

The first practical goal of our work is to produce a suitable formalised electronic version in which all the detailed information contained in the dictionary is encoded following the TEI (Text Encoding Initiative) guidelines [SPE 94]. The content of this electronic version updates the original EH according to the standard forms and the "rules" of the Basque Academy in order to ensure that all entries, sub-entries, definitions and example texts are in standard Basque. All this information will be integrated with the original version, and will be made explicit in the mark-up so that it can be readily identified, used or searched for. This formalised version of EH is a valuable lexical resource for NLP applications as well as the basis of the electronic EH (*e*EH). We aim to improve the facilities of an existing electronic version of EH in CD-ROM, because it is just the PDF version of the printed dictionary where linking facilities are only implemented for synonyms. For that reason, we studied the characteristics of the previous version and proposed a new representation and functionality. The use of these resources for general exploitation requires having a complete and accurate representation of them.

Regarding the representation, one of our goals is to build the possibility of active data interchange with other related NLP applications and contribute in the modernisation of Basque. That is why we have marked up the Machine Readable Dictionary (MRD) in SGML in compliance with the TEI guidelines. These guidelines comprise the only comprehensive attempt to provide a standard means which is able to encode machine readable texts.

When the conversion of the MRD to TEI is automatically performed, as in this case, the resulting output needs an exhaustive manual correction, specially if fine-grained analysis of the dictionary structure is carried out. This is the reason for making emphasis on this aspect. With respect to the new functionalities, we wanted to offer present-day users of EH the possibility to constrain their searches to particular sections of an entry as identified by mark-up.

1. The standardisation of EH with respect to the language is very important because we are so far involved in automatic processing of written Basque which is still under normalisation.

2. From the MRD to the TEI version dictionary

In the first step, the conversion of the MRD version of EH into a labelled structure was accomplished. The MRD version was intended for human rather than machine interpretation, since the lexicographer used a text-processor (Word Perfect, Word) to type the entries. As a consequence, we had to face a text file in which the only available codes were of typographic and lexicographic nature. In order to generate a structured representation of the information contained in the MRD, the following three main tasks were carried out : i) the analysis of the internal structure of the articles, ii) the specification of the grammar of entries that covered the general structure of the dictionary written as a Definite Clause Grammar (DCG) in Prolog, and iii) the conversion of the labelled structure which was encoded automatically[2] following the Text Encoding Initiative (TEI) guidelines, as explained in [ARR 96]. The TEI guidelines have been applied to the dictionary with considerable ease. Two benefits, then, that justify the application of a standard format are the reusability of the material and of the lexical sources it contains, and the possibility of using future utilities associated with standard formats.

2.1. *Dictionary grammar and automatic generation of the TEI encoded version*

The dictionary's structure is reflected in the parsing grammar. This is the grammar that the lexicographer had in mind when producing the dictionary. It was written as a Definite Clause Grammar (DCG) in Prolog. In order to illustrate the general structure of entries, we will use an auxiliary metalanguage as shown in figure 1 :

A | B	A or B, alternative nodes.
A B	A and B, A followed by B.
[A]	optional node.
ε	empty node.
a	terminal node (down case).

Figure 1. *Auxiliary metalanguage*

Figure 2 shows the syntax of the rules.

Element⇒ Element1 Element2 . . . ElementN.

Figure 2. *Syntax of the grammar rules*

2. The process of conversion of the labelled structure to TEI marked-up data has been performed by means of a PERL program that works on the analysis tree obtained after applying the DCG. This program follows a syntax-oriented strategy in order to create the adequate TEI mark-up corresponding to the different parts of the entries.

In figure 3 we have a part of the grammar[3] that describes the general structure of the entries.

Entry ⇒ Hdw [Relations] Category [date] [DefinitionExamples].
Hdw ⇒ [Homograph] [NonStdHdw | StdHdw].
Homograph ⇒ bh number eh.
NonStdHdw ⇒ gur bb hdw eb.
StdHdw⇒ bb hdw eb.
Category ⇒ [subc] Category.
Category ⇒ bi cat ei.
DefExamples ⇒ Def [Examples] DefExamples | ε.
Def ⇒ [SenseNumber][SenseGroup] def [Relations]
SenseNumber ⇒ bs number es.
SenseGroup ⇒ bs SenseGroup eg.
Relations ⇒ [SynRel | AntRel] Relations [Examples] | ε.
SynRel ⇒ bsy synonyms esy.
AntRel ⇒ ba antonyms ea.
Examples ⇒ bi examples ei.

Figure 3. *Simplified grammar of the general structure of entries.*

In general, an entry of EH includes : headword ; date ; variants ; part of speech ; abbreviations (style and usage labels, field labels, etc.) ; definition ; relations ; scienti-

3. Terminal nodes are labelled by "Btag" and "Etag" to mark their beginning and end respectively :

A The tags corresponding to lexicographic codes :

bd/ed : beginning and ending of the date.
bs/es : beginning and ending of the sense number.
bss/ess : beginning and ending of the subsense code.
bsg/esg : beginning and ending of the general sense group code.
bsy/esy : beginning and ending of the synonym relation code.
ba/ea : beginning and ending of the antonym relation code.
bh/eh : beginning and ending of the homograph identifier.

B The tags corresponding to typographic codes :

bb/eb : beginning and ending of bold.
bi/ei : beginning and ending of italics.

C Other tags :

bc/ec : beginning and ending of capital letters corresponding to subentries.
be/ee : beginning and ending of each entry.
gur : identifier of a non-standard entry.

```
<!element eh - - (entry)+>
<!element entry - - (hom|form|gramgrp|usg|def|xr|eg|sense|
                     note|re)+>
<!attlist entry type CDATA #IMPLIED>
<!element re - - (form|gramgrp|usg|def|xr|eg|sense|note)+>
<!element hom - - (form|gramgrp|usg|def|xr|eg|sense|note|re)+>
<!attlist hom  N CDATA #REQUIRED>
<!element sense - - (eg|def|usg|sense|xr|
                     gramgrp|form|note|re)+>
<!attlist sense N CDATA #IMPLIED>
<!attlist sense type CDATA #IMPLIED>
<!element form - - (orth|pron|usg|form|note)+>
<!attlist form type CDATA #IMPLIED>
<!element pron - - (#PCDATA)+>
<!element orth - - (hi|note|#PCDATA)+>
<!element gramgrp - - (pos|subc|number|usg)+>
<!element number - - (hi|#PCDATA)+>
<!element pos - - (#PCDATA)+>
<!element subc - - (hi|#PCDATA)+>
<!element usg - - (hi|note|usg|#PCDATA)+>
<!attlist usg type CDATA #REQUIRED>
<!element def - - (hi|#PCDATA)+>
<!element eg - - (q|xr|usg)+> }
<!element q - - (hi|note|usg|#PCDATA)+>
<!element xr - - (lbl|ref|hi1|#PCDATA)+>
<!attlist xr type CDATA #REQUIRED>
<!element lbl - - (ref|#PCDATA)+>
<!element ref - - (hi|#PCDATA)+>
<!element hi - - (hi|#PCDATA)+>
<!attlist hi rend CDATA #REQUIRED>
<!element note - - (hi|note|#PCDATA)+>
<!attlist note type CDATA #REQUIRED>
```

Figure 4. *DTD for the EH.*

fic names ; examples ; subentries, and grammatical information. All the data are given implicitly or explicitly in the hierarchical structure of the dictionary articles. The articles are structurally complex and present some problems that must be treated when analysing and interpreting them.

As a result of this conversion process, we recognised the structure of 98,49% of the entries with all the information they contained, giving an error rate of 3%. There were some errors related to the date and to some grammatical codes, but the parts of speech, definitions, examples, and so on were, in most cases, correctly recognised.

2.2. *DTD structure of the Euskal Hiztegia dictionary*

As explained above, the MRD version of EH has been translated into a collection of TEI-conformant SGML files. The TEI proposes a wide set of elements for practically all kinds of dictionaries (monolingual and multilingual) in electronic form but EH only needs a relatively small subset of them. In figure 4, we can see the DTD for EH[4] that is central to ensure the consistency and adequacy of the information contained in the dictionary.

3. Methodology for the manual correction of the TEI version

The method used for the manual correction of the generated TEI version guarantees and assesses the correctness, completeness, and adequacy of the information contained in the dictionary. The emphasis of this method relies on two points :

– The need to establish the criteria for checking the generated version.

– The need to perform a throughout quality check of the data produced, with special incidence on the entry by entry review.

3.1. *Establishment of the criteria for checking the generated TEI version*

Previous to the entry by entry review, a set of criteria for checking and correcting the generated TEI version was fixed. These criteria establish how to encode any element or part of an article ; their main objective is to guide the lexicographer who is manually reviewing the automatically produced version. When designing the criteria, all the aspects related to the structure of the information and the content of the dictionary were taken into account. In most of the cases, the features of EH could be readily squeezed into the TEI dictionary model without misrepresenting the source text. But in some cases, we were forced to make some modifications to the standard scheme, by adding new values for some attributes where nothing appropriate was found in the existing TEI tagsets. We will briefly present some examples of these modifications in the following section.

Regarding the TEI mark-up, we decided not to take into account the punctuation marks inside some elements or the brackets in the synonym tags. We also made the decision to write some specific abbreviations in the same font type, using always the same capitalisation schema.

All in all, in order to make linguists' task easier, we used an emacs-based interface that permits to browse, edit, and check the correctness of the syntax of the SGML tags according to the TEI DTD (Document Type Definition). Linguists also had on-line access to the EH and the criteria for checking and correcting the entries. The project

4. Of course, the reduced version here presented is TEI-conformant.

has involved five persons in the design task and supervision committee, one person for linguistic coordination, two for linguistic work and two more for computer support.

In order to ensure the consistency and completeness of the entries, the linguist coordinator appointed ordinary meetings with the author of the dictionary. In fact, we also included those corrections that were already manually done by the lexicographer on the printed dictionary in the correction process.

3.2. *Entry by entry review*

Once the automatically generated TEI version was finished, we saw the need of an exhaustive manual correction in order to ensure the adequacy of the generated mark-up. To do so, we looked up in the dictionary entries to see to what extent the tags needed to be adapted to the shape of the entries. For instance, there are some entries containing common expressions such as "*Odolak ur bihurtu zaizkio :* erabat beldurtu da." ("His blood turned into water : he was completely scared."), where the meaning of the common expression is also explained. Since these particular cases are not foreseen by TEI gidelines, we decided to define a new value for the *type* attribute of the *usg* element for this type of example (`<usg type='common_expression_meaning'>`).

However, that was not the only point we had to take into account. Even though the generated version seemed to be quite satisfactory at first sight, there were some elements that were not properly encoded due to different factors. In most cases, these unsatisfactory analyses were caused by typographic errors or inconsistencies of the source MRD. For instance, when in the original version the lexicographer wrote a definition in italics (instead of in a regular font), this definition is marked as an example. Furthermore, there are some entries that were not correctly analysed due to their special nature. These entries contain very detailed information and constitute a particular kind of entries that were not covered by the automatic analysis. Therefore, in order to get a fine-grained analysis of the dictionary, we have combined the automatic step with manual or semi-automatic lexicographic labour.

```
<def> <hi rend=italic>berezk.</hi> egur bezala erabiltzen dena.</def>
```

Example 1: Changing the default font type. The definition part would stay (in English) as :"*Especially* used as wood."

Besides, there was a different type of correction concerned with the original version of the dictionary and the decisions we wanted to take on it. This is the case of typing errors (*tttipia* → *ttipia* ("small"), *biurtu* → *bihurtu* ("to convert") ...) or the decisions to make on how we wanted the final version to look like. Some examples follow :

1) We defined a default font type to each tag so that we only had to mark specific cases when they were inside tags which already had a default font type (see example 1).

2) Normalized form for abbreviations : for instance, the different abbreviations for "common use" as preference level (*ohi., Ohik., ohi, ...*) are reduced to one (*Ohi*) ; in the case of the abbreviations for the part of speech "noun" (*ize., iz., iz, ize, ...*) we use only one (*Ize.*), etc.

3) Except in some particular elements (definitions, examples, dates, etc.), the punctuation marks are removed.

4. Standardisation of the dictionary with respect to the language

Bearing in mind that since the last edition[5] of EH the Basque Academy has made new decisions about the standard forms of some words, we based our job on the Basque Academy Dictionary [EUS 00] and the "rules" of the Basque Academy in order to ensure that all the entries, subentries, definitions, and example texts are written in standard Basque.

The standardisation with respect to the language is always important but, in our case, it is crucial because we are still involved in a process of normalisation for written Basque. The process has been performed in two steps. At a first stage, we have standardised the entries and subentries according to the Basque Academy "rules", integrating the information contained in the Basque Academy Dictionary. We detected two different cases :

– Those headwords that EH considers standard but the Basque Academy does not. All these entries have now a special *type='dead'* attribute in the headword, telling us that the entry is not anymore standard. In this case, there are two kinds of entries, some containing definitions or examples and others just having a variant meaning :

a) In the case of the entries offering a definition or an example, we have inserted a *note* element with type *ATA*[6] to indicate that the Basque Academy does not approve the given headword (see example 2).

```
<form type='dead'><orth>apart</orth></form>
   <note type='ATA'>apart* e. aparte</note>
```

Example 2: Non-standard entry *apart*. According to the Basque Academy, its standard form is *aparte* ("out of the way")

b) When the entry has no definition or example, the non-standard entry itself indicates its standard equivalent. In this case, we made use of a cross-referencing code

5. The first edition of EH (called *Hauta-Lanerako Euskal Hiztegia*) was published in 9 fascicles between 1984 and 1995 by Ibon Sarasola. The second edition in 1996 is a revised and updated version, which is also in CD-ROM. One of the main aims of this dictionary was to clarify the doubts of anyone needing information about the standard forms of written Basque. In order to contribute in the standardisation process, the EH includes non-standard forms and variants of the standard form entries.

6. According To the Academy.

(*Ik.*) in order to refer directly to the standard form. That standard entry will provide us with the necessary definitions or examples, if any (see example 3).

```
<form type='dead'><orth>artegai</orth></form>
   <xr type='std.'><lb1>Ik.</lb1><ref>artelan</ref></xr>
```

Example 3: Standard form for word *artegai* ("work of art")

c) Regarding those headwords accepted as standard by the Basque Academy but not by EH, we simply standardised the spelling of the non-standard entries.

In addition to all this, there are some sense distinctions of the Basque Academy Dictionary that do not match with those of EH. As a consequence, we have also coded these mismatches in order to bring up to date the correct use of each of the senses in the dictionary. In example 4, we can see the use of the word *izenorde* as defined by the Basque Academy Dictionary.

izenorde 1 'ezizena'. **2*** *e.* izenordain.

Example 4: Entry for the word *izenorde* in the Basque Academy Dictionary. This word has two senses. The first one refers to *ezizena* ("nickname, pseudonym") but the use of the word *izenorde* for the second sense (meaning "pronoun") is declared non-standard (note the asterisk) : *izenordain* is proposed instead as the standard form for it.

The Academy states that, for the second sense of that word, we should use the form *izenordain* instead of *izenorde*. Therefore, we encoded this entry in the TEI version of EH as shown in example 5.

```
<sense n='2' type='dead'>
   <note type='ATA'>izenorde* e. izenordain</note>
```

Example 5: Encoded TEI-version for entry *izenorde*.

Once we finished the standardisation of the entries, subentries, and the definition fields, we began working on the correction of each entry's examples and definitions. By means of an automatic spelling checker program, *Xuxen* [ADU 97], we were able to detect the misspellings present in the texts. Therefore, our main task was to correct all the mistakes, always according to the Basque Academy rules. Most of the corrections had to do with spelling mistakes such as missing or additional characters, joined or separated words, etc. Apart from the common spelling mistakes, we found many examples taken either from old Basque or from different dialects. In most of the cases, it was impossible to guess the original form of the mistaken word. Thus, we decided not to modify the example texts, on the one hand, in order to keep their original rhythm, tone and style, and on the other hand, due to the fact that neither the Basque Academy nor the *Xuxen* speller provided us with the corresponding standard forms.

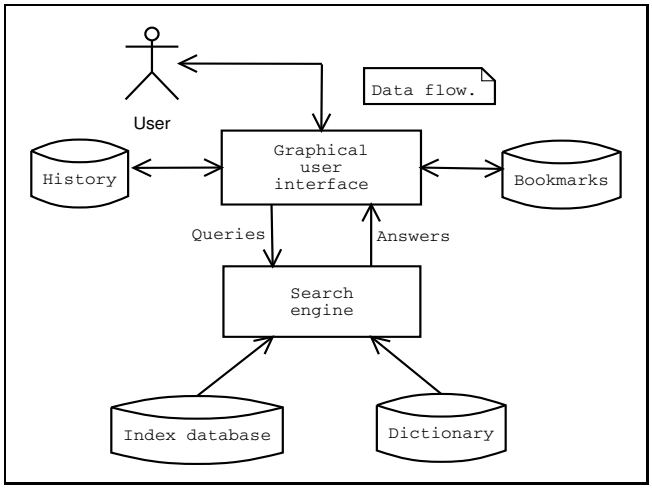

Figure 5. *General architecture of the* e*EH application.*

5. Search tools and query interface

At this stage of the project, we have a complete lexical repository which is manually corrected. Such a database is a very important resource for Basque NLP research. We have designed and implemented an enhanced on-line search tool over the dictionary, adding new facilities and functions to the classic dictionary access. The application is designed to try to satisfy the necesity of two main user groups. On the one hand ordinary users who need an easy-to-use and to-the-point tool to access a dictionary but few extra features, and on the other hand higher-level language students and users, such as writers, translators, journalists, and linguists. They usually need many advanced features, some of which we have tried to satisfy with this application, including search capabilities based on the occurrence of a word in a given context, i.e., the use of the dictionary as a searchable text corpus.

5.1. *General architecture of the application*

Figure 5 shows the general architecture of the consultation system. Two main modules have been identified :

– The **Graphical User Interface**, which interacts with the user. It gets the queries s/he poses, sends it to the search engine, gets the result of the search and renders the entries according to a pre-established stylesheet. It maintains the history of the queries posed so far. The user has also the possibility to store the most interesting answers in a bookmark-like repository.

– The **Search Engine**, which actually performs the search. It needs the index database and the dictionary itself. To answer a given query, the application uses only the

```
<entry id='A6'>
  <form><orth>abade</orth></form>
  <GramGrp><pos id='A6S0'>iz.</pos></GramGrp>
  <usg type='time'>*1562, ~1620</usg>
  <sense n='1' id='A6S1'>
    <def id='A6S1D1'>Apaiza.</def>
    <usg type='geo'>Bizk.</usg>
    <eg id='A6S1A1'><q>Abade jauna.</q></eg>
    <eg id='A6S1A2'><q><usg type='esr_zah.'>Nolako
    elizalde, halako abade.Abadearen lapikoa, txikia baina
    gozoa.</usg></q></eg>
    <sense n='n1' id='A6S2'>
      <eg id='A6S2A3'><q>Abade egin.</q><xr type='syn'>
      <lbl>Ik.</lbl><ref>abadetu</ref></xr></eg>
    </sense>
  </sense>
  <sense n='2' id='A6S3'>
  <usg type='time'>1635</usg>
  <def id='A6S3D1'>Gizonezkoentzako monasterio bateko
     burua, apaizteko esku duena.</def>
  <xr type='syn'><lbl>Ik.</lbl><ref>abadesa</ref></xr>
  <eg id='A6S3A1'><q>Paulo abade zahar hura.
     Leireko abadea.</q></eg>
  </sense>
</entry>
```

Figure 6. *TEI-conformant entry for* abade *with logical IDs included*

index database, performing boolean operations among large sets of indices [WAR 92], and gets the corresponding indices of the entries which satisfy the query. In a final step, the search engine retrieves the entries from a serialized binary image of the dictionary. The index database and the dictionary will be explained in detail below.

5.2. *Indexing the dictionary*

In order to answer quickly the queries posed on the *e*EH by the user, the dictionary must be indexed ; thus, an indexed database of the dictionary itself has been built using the "inverted files" technique [RIB 99].

Every entry, related entry, sense, definition, and example of the dictionary receives a unique *logical identifier*. For instance, in figure 6, we can see the SGML

TEI-conformant marked entry corresponding to the word *abade* ("abbot") with the logical identifiers included[7].

Furthermore, every word in a definition or example, receives a unique logical ID, formed concatenating the Definition/Example logical ID with the relative position of the word. Thus, each word in the first example of the entry *abade* (marked with a square box in figure 6), "*Abade jauna*" ("Lord abbot") will have the identifiers showed in example 6. Note that such a logical ID gives the application the possibility for locating any word in the dictionary. For instance, the A6S1A1K1 index for *jauna* ("lord") in the example shows us that this is the second word of the first example of the first sense of the 6th entry starting with "A".

abade	⇒	A6S1A1K0
jauna	⇒	A6S1A1K1

Example 6: Indexing words in definitions and examples

We constructed an index database, linking each word form with a list of all occurrences of this particular form in the dictionary. Apart from this, we also lemmatise all the words by means of a tagger/lemmatiser [ADU 96], linking each lemma with its corresponding word forms. This way, we give the application the possibility of querying not only about word forms, but also about lemmas.

The index database itself consists of several independent but related index files :

– **Form Index :** For every word form in the dictionary, this index stores a unique ID and associates it to the list of lemma IDs corresponding to this form[8].

– **Lemma Index :** For every lemma, there is a unique ID, a list of the IDs corresponding to the word-forms which share this lemma and an optional entry logical ID list[9].

– **Category Index :** Every possible POS is linked to all the dictionary senses which share them.

– **Entry Index :** For every entry and related entry, the file keeps a unique logical ID.

– **Position Index :** It links every form ID with all its occurrences in the dictionary, via logical IDs. This index is actually split into two different files, storing the position of the definition's words, and the example's words, respectively.

Finally, the **Physical Position Index** relates every logical ID to the physical position of this entry in the dictionary file. We have decided to store the TEI version of EH

7. The figure is only for illustrating the way logical indices are attached to each element. Actually, these ID's are not stored in the SGML documents, but in the index database.
8. Because of lexical ambiguity, a word form may have different lemmas.
9. Note that the dictionary entry headwords are lemmas themselves.

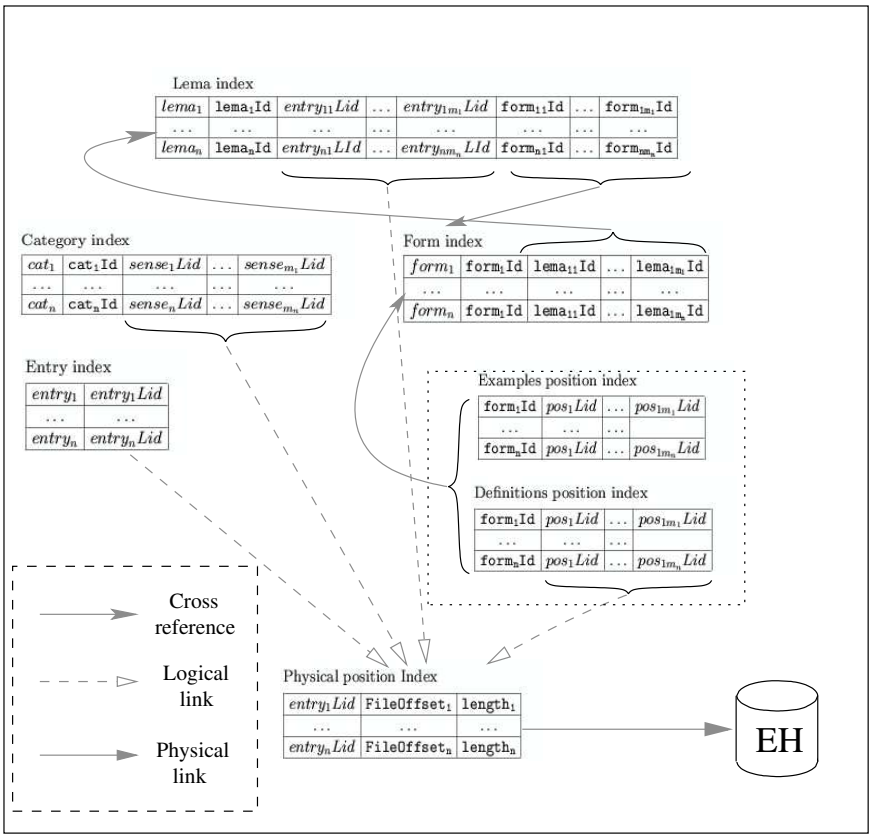

Figure 7. e*EH Index system.*

in a binary format, i.e., serialising the memory representation of the parsed dictionary. In order to accomplish this task, a collection of abstract data types has been defined based on the elements present in the DTD.

Some related works (i.e. the Anglo-Norman Dictionary[10]) also deal with the conversion of legacy dictionary data into XML format ; however, they recommend not to change the dictionary format after the XML conversion and to consider this version as the only lexical source : XML related tools should be used to access the dictionary entries. We have not adopted this approach, mainly, because of the following two reasons :

– If the actually stored dictionary is a collection of SGML/XML marked text files, the application will have to parse every entry whenever an access to this entry is required. As the parsing of the entries is not a trivial task, this would be a time-consuming

10. http ://and.lexilog.net :8090//techbriefl.html

```
Query ⇒ UniqueFieldQuery Query | ϵ
UniqueFieldQuery ⇒ entry = String ; |
                   example = ConstraintOnExample ; |
                   definition = ConstraintOnDefinition ; |
                   pos = ConstraintOnPos ;
ConstraintOnExample ⇒ QueryOnWords
ConstraintOnDefinition ⇒ QueryOnWords
QueryOnWords ⇒ ContinuousWordsQuery |
               UncontinuousWordsQuery |
               ε
ContinuousWordsQuery ⇒ " UncontinuousWordsQuery "
UncontinuousWordsQuery ⇒ EhRegex UncontinuousWordsQuery |
                         EhRegex + UncontinuousWordsQuery
ConstraintOnPos ⇒ Pos
EhRegex ⇒ (asterisk?character+) + asterisk?plus?
```

Figure 8. *Grammar for the queries in* e*EH. In the last line of the figure, " ?" represents that the preceding symbol may be repeated 0 or 1 times and, "+" that the preceding symbol may appear 1 or n times. As it is expressed in the rule, a reference to a word in the query may begin or not with "*" and be followed by, at least, a character. This sequence may be repeated as many times as needed. The last characters may be "*+", or just "*" or "+".*

trade-off in our system.

– Due to copyright issues, the dictionary will be stored in an encrypted way. Therefore, we will actually have to change the dictionary format anyway.

Figure 7 shows the internal structure of these index files and the relations between them.

5.3. *Language for queries*

All the possible queries –simple and advanced– that the system can handle are expressed by means of the grammar shown in figure 8. We would like to remark the two main possibilities when posing a query :

1) Search for complete phrases (ContinuousWordsQuery rule). It is represented by enclosing word forms or lemmas of the phrase in quotation marks. They will appear together in all results exactly as you have entered them.

2) Search for different words/lemmas occurring anywhere in a given field (definitions or examples), although not necessarily in the same order.

In all the cases, a word form can be expressed using regular expressions ; we can ask for all the forms containing the *tegi*[11] suffix by using the expression "*tegi". The "+" character is used to ask about lemmas. For instance, a query "zakur+" will return all the forms which share the lemma *zakur* ("dog") , such as *zakurrarekin* ("with the dog"), *zakurrari* ("to the dog") etc.

6. Examples of use of the application.

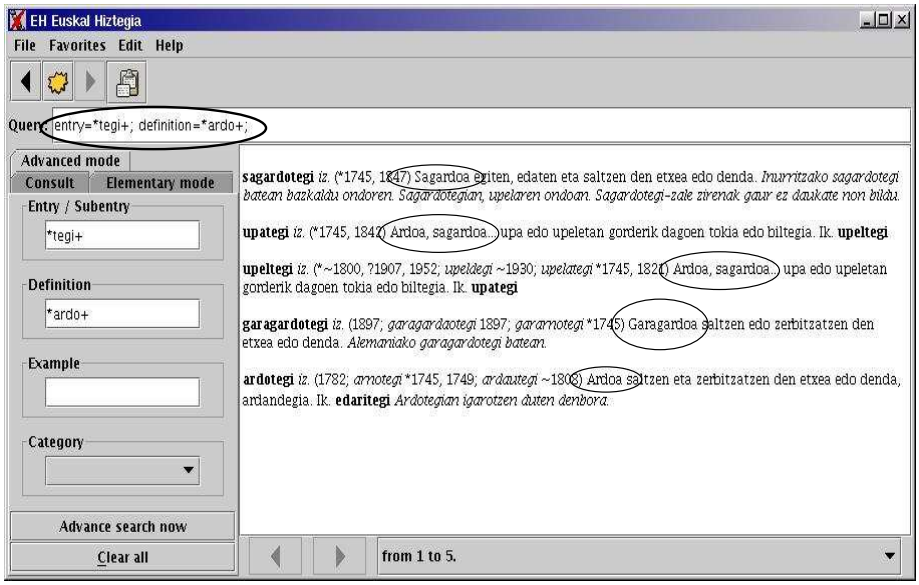

Figure 9. *Graphical Interface for querying the* e*EH.*

As stated before, the application is designed to try to satisfy the need of two main user groups : ordinary users and higher-level users. As none of them is supposed to be computer masters, a GUI (Graphical User Interface) has been designed and implemented[12]. The interface must be easy-to-use and intuitive, and this meant that it was not desirable to make users learn a complex query language. In order to fulfil these requirements, we designed a split GUI, with a left panel for queries and a right one for result display. The left panel is itself divided into three tabs, each providing an incremental set of capabilities (from traditional dictionary lookup to more complex searches). The most complex searches offer the possibility to combine conditions on each of the searchable fields. On the other hand, the right panel shows the entries that satisfy the query posed in a friendly way.

11. Suffix to denote "place".
12. The GUI is still in beta state.

Full hypertext was also added to the application in order to make it possible to jump from "almost" any word in an entry to its definition with just a double click. The few exceptions to this possibility are the non-regular entries of the dictionary (foreign words, person/place names, etc.). To avoid the side effect of losing a possibly valuable search result when clicking or making a new search, an Internet browser-inspired back-and-forward capability was added. Also, a "favourites" folder was conceived to group search queries and results valuable to the user. In case the searched word has more than one possible lemmas, a dialogue box will permit users to select the desired one. In figure 9 we can see a partial view of the interface.

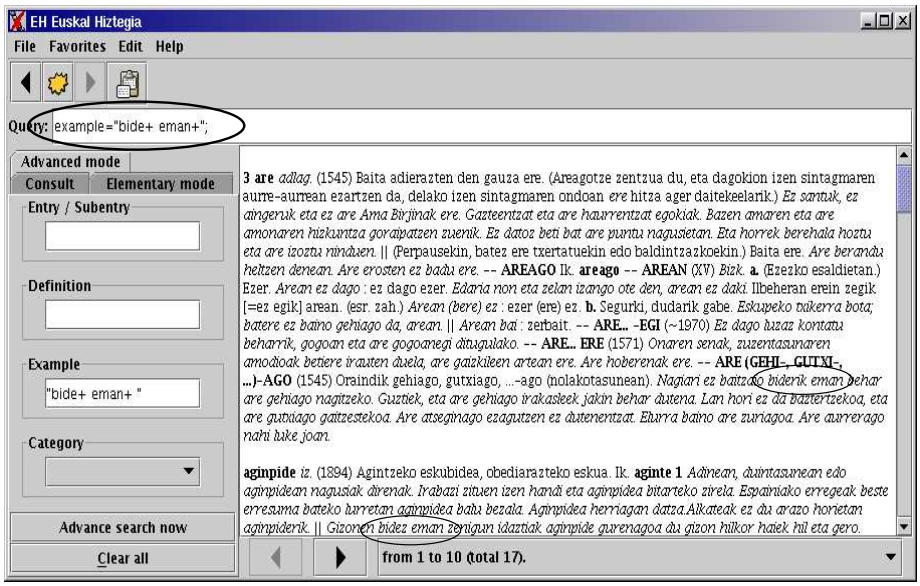

Figure 10. *Graphical Interface for querying the* e*EH (2).*

In figure 9 we can see the answer to the advanced query "give me the dictionary entries/subentries corresponding to words whose lemmas end with the suffix *tegi* and, at the same time, contain in their definition words whose lemmas end in *ardo* ("wine as an alcoholic drink made from fruit, plants, etc."). Figure 10 shows another example of an advanced query :"give me the dictionary entries/subentries which contains in their example section the sequence composed by words whose lemmas are *bide*, followed by words whose lemmas are *eman*". In this case, the user is interested in looking for occurrences of the expression *bide eman* ("to make possible"). As an answer for this posed query, the system returns 17 entries/subentries. Only the first ten ones are shown, although the user can also visualise the rest of the answers. In figure 11, we restrict the previous query only for nouns.

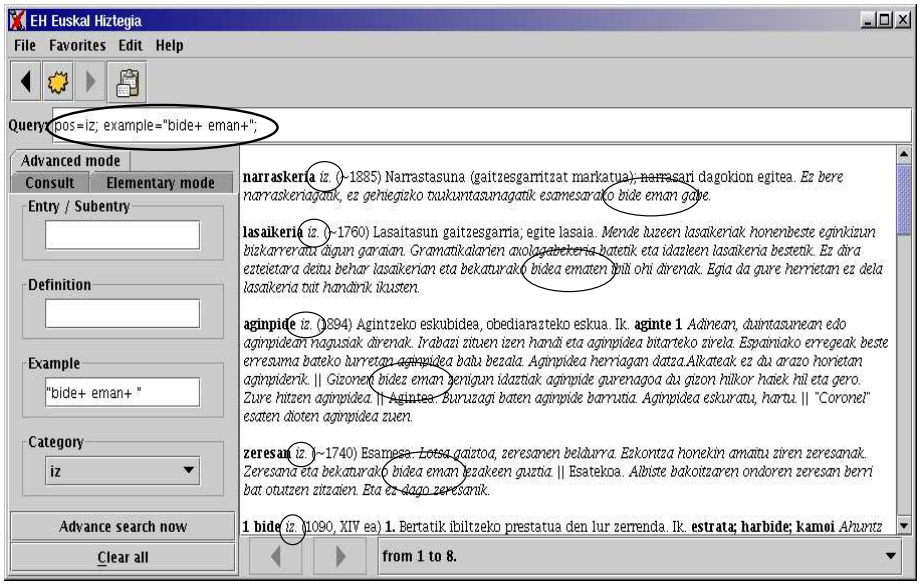

Figure 11. *Graphical Interface for querying the* e*EH (3).*

7. Future work and conclusions

In this paper we have presented the procedures involved in the development of an advanced electronic version of the EH dictionary, called *e*EH. We first translated the MRD version of EH into a TEI conformant format using automatic methods, and later we have performed an additional manual or semiautomatic review. The translation of the EH from the MRD into the TEI version is an essential task in order to develop a more advanced electronic dictionary. In addition, the inconsistencies we have found through the work of adaptation reveal the lack of a systematic lexicographic work and the need of more consistent criteria when editing dictionaries. From this point of view, the use of the TEI version of the EH repository is a further step in the design of such an edition environment for updating and improving the dictionary itself. As a result of this work, we can use the information contained in this lexical resource as a basis in future applications and research. Besides, we have designed and implemented an enhanced on-line search tool over the dictionary, adding new facilities and functions to the typical dictionary access. The application has been designed in order to satisfy the need of two main user groups, namely, ordinary users, who need an easy-to-use and to-the-point tool to access a dictionary but few extra features, and higher-level language students and users, such as writers, translators, journalists, and linguists. This second group has a need for many advanced features, some of which we have tried to satisfy with this program.

A fully working prototype of *e*EH has been implemented. The structured version of the dictionary has been already used in the production of a dictionary of synonyms and in the construction of a Basque semantic net. In the future, we will improve this prototype by adding and integrating it in a text editor. Besides, we are considering the possibility to extend the index system considering grammatical information associated to the words in definitions and examples. The extended index system will allow us to query not only about word forms/lemmas, but also to combine them with additional information (e.g. POS values). The extended index system is the next step towards the integration of *e*EH in general NLP applications.

8. Acknowledgements

This research was partially funded by the European Commission under the Feder program (project 2FD1997-1503) and the University of the Basque Country (EX-19-1999).

9. Bibliographie

[ADU 96] ADURIZ I., ALDEZABAL I., ALEGRIA I., ARTOLA X., EZEIZA N., URIZAR R., « EUSLEM : A Lemmatiser/Tagger for Basque », *Proc. Of EURALEX'96*, Göteborg, (Sweden), 1996, p. 17-26.

[ADU 97] ADURIZ I., ALDEZABAL I., ALEGRIA I., ARTOLA X., EZEIZA N., SARASOLA K., URKIA M., « A spelling corrector for Basque based on morphology », *Literary & Linguistic Computing. Oxford University Press. Oxford*, vol. 12, n° 1, 1997.

[ARR 96] ARRIOLA J. M., SOROA A., « Lexical Information Extraction for Basque », *CLIM'96*, Montreal, 1996.

[EUS 00] EUSKALTZAINDIA, *Hiztegi Batua. 45. Liburukia*, Euskaltzaindia, 2000.

[RIB 99] RIBEIRO-NETO B., BAEZA-YATES R., *Modern Information Retrieval*, Addison Wesley, 1999.

[SAR 96] SARASOLA I., *Euskal Hiztegia*, Kutxa Fundazioa, Donostia, 1996.

[SPE 94] SPERBERG-MCQUEEN C. M., BURNARD L., *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*, Chicago & Oxford, 1994.

[WAR 92] WARTIK S., « Boolean Operations », FRAKES W., BAEZA-YATES R., Eds., *Information Retrieval. Data Structures & Algorithms*, Prentice Hall PTR, 1992.