Agnès TUTIN, Unité de Recherche Associée SILEX, CNRS & Université de Lille III,
Jean VÉRONIS, Laboratoire Parole et Langage, CNRS & Université de Provence

Electronic Dictionary Encoding: Customizing the TEI Guidelines

Abstract

The Text Encoding Initiative (TEI) *Guidelines* propose an encoding scheme that is applicable to a large variety of dictionaries. This paper takes a widely-used and simple French dictionary (*Petit Larousse Illustré*) as an example to describe some of the problems that arise when using the TEI Document Type Definition (DTD), particularly from an editorial standpoint, where it is necessary to retain the presentational features of a text. It also discusses general issues in DTD customizing and compatibility.

Keywords: Dictionaries, encoding, TEI, SGML, customization

1. Introduction[1]

The Text Encoding Initiative (TEI) *Guidelines* (Sperberg-McQueen & Burnard, 1994) propose an encoding scheme that is applicable to a wide variety of dictionaries (Chapter 12, *Print Dictionaries*[2]). Other attempts have been made to produce a formal description of the general structure of dictionaries (e.g. Danlex, 1987; Hausmann & Wiegand, 1989), but the TEI can be accredited with having reached a consensus in the international community and having developed a Document Type Definition (DTD) for encoding in the SGML language (ISO, 1986).

The present paper builds largely on a previous study by Ide & Véronis (1995), which points out the unavoidable shortcomings that result from the very high level of generality of the TEI DTD. Ide & Véronis (1995) describe the inevitable conflict between *generality* and *precision* that arose within the TEI: in an attempt to model the largest possible number of documents, the TEI had to give up tight modeling of each document (sub)type. As a result, the TEI DTD grossly *overgenerates*, i.e., it does not precisely delineate the universe of accepted documents in a given application. For example, the dictionary-specific DTD of the TEI is capable of representing not only many structures that do not exist in any dictionary but also structures which, while occurring in different dictionaries, never simultaneously appear in the same one.

In many applications, then, the TEI DTD must be customized, especially for tasks like keyboard input, printing, and uptranslating from pre-existing computerized formats. In particular, Ide & Véronis (1995) point out that dictionaries can be seen from different views, depending on individual needs and users. More specifically, they describe:

- the *editorial view*, which is based on meta-lexicographic analysis and must allow for the reconstruction of the typographical form of the text, except for variations in page layout;
- the *lexical view*, which totally disregards format and only takes into account the informative and linguistic content of the lexical entry (a dictionary is thus viewed as a lexical database and not a text).

Although it is feasible in theory to encode both views in the same document (see TEI Chapter 12), this turns out to be extremely tedious and complicated in practice. It is more realistic to develop conversion systems for switching from one view to the other.

In this paper we describe some of the problems that arise when the TEI DTD is used directly, particularly from an editorial standpoint where the presentational features of a text must be retained, by taking a widely-used, simple French dictionary (*Petit Larousse Illustré*, hereafter PLI) as an example. We also discuss general issues in DTD customizing and compatibility. In particular, we claim that the customizing process should abide by precise rules in order to ensure compatibility with the original TEI DTD, and if possible, automatic convertibility.

2. Limitations of the TEI DTD

Like all undertakings of this scale, the TEI DTD has a certain number of shortcomings or problems which affect dictionaries as well as other types of documents. Some of these are simple omissions and can easily be remedied; others require further analysis of the components involved and the relationships between them. For a lack of space, we will look solely at customizing problems.

Depending on whether we take the editorial view (where the typographical form must be recoverable) or the lexical view (where the main information classes of an entry are represented regardless of their form), the kind of customizing that needs to be done is not the same. Since the TEI DTD is largely based on the lexical database model proposed by Ide, Le Maitre & Véronis, 1993, it is much easier to adapt to the (more abstract) lexical view than to the editorial view, where a systematic mapping between DTD elements and their typographical renditions must be defined.

Moreover, it has often been said that the TEI DTD is complicated and difficult to understand. The use of *parameter entities*, which attempt to simulate a class system via a purely syntactic mechanism, makes reading the DTD particularly cumbersome. It is thus extremely difficult for anyone who reads the *Guidelines* to determine what a content model actually contains, even one as basic as the paragraph model (*%paraContent;*). Although the modularity that SGML lacks is cleverly simulated by the TEI DTD via the inclusion of sub-DTDs (*tagsets*), this method has unforeseen and undesirable side effects.[3] In addition, direct input and manual uptranslation can profit considerably from the use of editors like *Author/Editor* or *Grif*[4], which assist and check the encoder's work by proposing a menu that lists the legal insertions at each point in a document. However, a large number of choices are proposed to the encoder, even though most of them are never valid in dictionaries, and this makes this function quite impractical. These considerations that led the editors of the TEI to devise a simplified DTD, *TEILite* (Burnard & Sperberg-McQueen, 1995), capable of encoding most ordinary texts. The simplified DTD only retains the most common tags and does not have to resort to syntactic mechanisms that would lessen its readability. However, *TEILite* is not suitable for encoding dictionaries, since it has none of the dictionary-specific tags. A simplification of the TEI *Guidelines* specifically aimed at dictionaries would therefore be welcome.

Finally, the SGML editors usually allow the user to validate a document. Validation is extremely important for the encoder, since it helps detect errors. However, in addition to tags that never appear in dictionaries as mentioned above, the TEI's dictionary DTD allows most

tags anywhere although they are likely to be valid only at a given point in a particular dictionary. In the PLI, for example, the etymology is always located at the beginning of an entry, after the synchronic identification fields, whereas the DTD allows it to occur anywhere in the entry. This drawback, which affects nearly all tags, substantially lowers the efficiency of the validation process and must be offset by the addition of position or repeatability constraints to the DTD, and by renaming (or "specializing") certain elements according to their context. It is however undesirable to completely give up the standardization offered by the TEI. The best solution would be to have both a "proprietary" DTD designed for input ease and faithfulness to a given typographical rendition, and the TEI DTD for the purposes of interchange and use of standard tools. Reversible mechanisms can be designed for converting to and from either tagset.

3. Customizing Principles

The DTD *Guidelines* are very loose when it comes to modifying the DTD (Chapter 28, *Conformance*) since one can delete practically all of the original elements and content models and replace them with new ones, while still remaining "*TEI-conformant*". It seems to us, however, that DTD customizing should be done in accordance with certain predefined principles, so as to avoid the proliferation of incompatible DTDs that would revert back to the pre-TEI era. These principles must guarantee that all documents have certain properties, and above all, that they can be converted to the original TEI DTD. Convertibility to a shared DTD is essential, in particular, because standard software can be used. Furthermore, for the specific case of dictionaries, the editorial view does not preclude taking a later lexical view of the dictionary created, i.e., using it as a database. But once the DTD is modified for the editorial view, it is not immediately suitable for the lexical view.

To our knowledge, the only notion used so far in DTD modification is *tag subsets*, in the set theoretical sense. But this notion is too limited, since the selection of a tag subset does not account for what happens to the content models, and it does not provide for simple transformations (often reversible) like name changes, element moves, etc.

Three preliminary definitions are needed here[5] (technical details are omitted):
1. Let D be a DTD. Then the *universe of D documents* (or simply the *universe of D*) refers to the set $\mathcal{U}(D)$ of documents accepted by D.
2. Let D and D' be two DTDs. Then we say that D *subsumes* D' (or D' is *conformant to D*) if $\mathcal{U}(D') \supseteq \mathcal{U}(D)$, i.e., all documents accepted by D' are also accepted by D.[6]
3. Let D and D' be two DTDs. Then we say that D *subsumes D' modulo φ* (or that D' is *conformant to D modulo φ*) if there exists an application φ such that for any document instance $x \in \mathcal{U}(D')$, $\varphi(x) \in \mathcal{U}(D)$.

A document instance x can be regarded as a tree whose leaves are character strings (i.e. #PCDATA) and whose nodes are "decorated" with tags and attribute-value pairs. The application φ is thus a tree transformation that affects node decoration, tree configuration, leaf content, or any combination thereof. If φ is a one-to-one mapping, then the transformation is *reversible,* i.e., there exists a transformation φ^{-1} that can reconstruct the original document instance.

There are various classes of transformations (tag renaming, adding or deleting grouping tags, conditional location changes, etc.). Some (reversible) examples are:

`<A type="t">abc</A>`	↔	`<B>abc</B>`
`<A type ="t"></A>`	↔	`<B>t</B>`

A language like SgmlQL (Harié *et al.*, 1996) performs such conversions automatically. It is clear that (i) convertibility to the TEI is guaranteed if the TEI DTD subsumes the modified DTD, modulo a given transformation, and (ii) it is preferable to have this transformation be reversible.

4. An example of customizing for an editorial view: the PLI

As mentioned above, the TEI DTD can be very easily customized for the lexical view. However, customization for the editorial view is more complex, and we will examine in this section the operations we had to perform for the PLI.

4.1. Lexical view and editorial view[7]

First of all, tagging can be done with a single element even when the fields convey the same type of information but have different distributions or typographical renditions. For example, the mention of the plural form exhibits two different kinds of presentation depending on the grammatical status of the entry (phrase and/or noun). Thus, in the PLI, the plural form of INTRA-UTÉRIN ('intra-uterine') occurs after the part of speech information, while for the adjective BANAL ('banal', 'commonplace'), the same information appears in the entry form.

BANAL, E, ALS adj. [...]	**INTRA-UTÉRIN, E** adj. (pl. *intra-utérins, ines*) [...]

Moreover, it is always possible to add information to an element using the many attributes offered by the TEI DTD.

On the other hand, the editorial view places more constraints on the user, since entries are not simply analyzed content-wise but also in terms of their form. Both the distribution and the rendition of each information field must be taken into account in designing the DTD. Rendition is not coded as such, but must be recoverable through a mapping of the DTD elements to their typographical rendition for printing and display.

In customizing the DTD for the PLI, we required that exact typographical rendition of entries be possible. Exact rendition can be achieved by any SGML editor[8] that associates a formal description to every element of the DTD. The following operations were performed:

- All characters with a significant text content were treated as tagged characters within an element, including sense numbers, which are difficult to generate automatically.
- Fields with different typographical renditions were considered as separate SGML elements. Even slight typographical differences generated a new element.
- Typographical characters used solely to separate fields (field delimiters or boundary markers), but conveying no other information, were not incorporated into the tagged

text. It was assumed that when an SGML editor like *Author/Editor* is available, these delimiters can be generated automatically as element "prefixes" or "suffixes".

- Elements had to be ordered and the [± optional] and [± repeatable] features of each one had to be marked.
- Whenever possible, implicit information was made explicit through attributes.

The association of a given format (style and/or boundary markers) to each element was thus the basis for element delineation.

4.2. Modifications of the TEI DTD

Let us now look more concretely at what operations we had to perform to customize the TEI DTD.

4.2.1. Order constraints and occurrence constraints

To facilitate input, an order had to be defined (a) among the main DTD elements (e.g., the <etym> field, which gives the complete etymology), and (b) within those elements. This type of constraint is indispensable for guiding the encoder through the tagging process. In conjunction with this, one must also specify whether an element is mandatory and/or repeatable (the TEI DTD considers almost all elements to be optional and repeatable). This operation was quite easily performed since the PLI entries are tightly structured. We never-theless noticed miscellaneous "floating" elements which could be included in many elements. For example, a grammatical information concerning the position of the adjective (s.v. INÉVITABLE "Avant le n." 'before the noun'), or the grammatical restriction on the subject (s.v. IMPORTER "[ne s'emploie qu'à l'inf. et aux 3è. pers]" 'is only used with the infinitive or with third person forms') can occur after the entry or at the beginning of the sense if the entry is polysemous.

INÉVITABLE adj. 1. Qu'on ne peut éviter ; fatal, inéluctable. 2. (Avant le n.) A qui ou à quoi l'on a forcément affaire ; que l'on ne peut éviter de subir. *L'inévitable raconteur d'histoires drôles des fins de banquets.*

IMPORTER v.i. et t. ind. *[à]* (it. *importare,* être d'importance) [ne s'emploie qu'à l'inf. et aux 3è pers.] [...]

The other "floating" elements are usage marks, lexical cross-references, emphasized phrases and encyclopedic comments.

4.2.2. Deleted elements

The Document Type Definition of the TEI is a general purpose DTD designed to cover a wide range of dictionaries of variable complexity in many languages. It also contains many general text tags (in the *core tagset*) which are useless in the vast majority of the dictionaries. Due to the fact that the PLI is a simple, monolingual dictionary, and due to the particular features of its language, most core tags (except paragraph marks and highlighting) and certain dictionary-specific elements were omitted. Deleted elements obviously do not cause problems for the conversion rules: a DTD in which optional elements are omitted is subsumed by the original DTD.

4.2.3. Added elements

As mentioned above, the TEI DTD is not detailed enough for the editorial view, so we had to introduce a number of new elements (See in Appendix A the list of elements created for the

PLI application). Most new elements were added to enable conformity with the editorial view. Two main other reasons were also at the origin of the creation of new elements. First, the frequency of informational fields has been taken into account so as to lighten the tagger's task. We sometimes added an element not because the TEI was unable to treat the given field, but because the treatment appeared tedious and cumbersome given the high frequency of the phenomenon. For example, nominal and adjectival inflections are processed in the TEI DTD (see *Guidelines*, section 12.3.1) with the help of the same elements as the entry lemma, **<orth>** and **<form>** with a specific attribute-value pair (*type="inflected"* on **<form>**). We thought it more convenient to introduce a specific element for a field which appears very frequently in romance language dictionaries (almost every adjectival entry and many nominal entries will contain such fields).[9] Secondly, the TEI tagset proposed for dictionaries sometimes appears insufficient to account for a relevant analysis enabling to isolate all the informational fields (even if some other TEI elements could be used to restitute the exact typographical rendition). For example, we chose to add a specific element **<phrase>** for idioms or collocations introduced within the definitional field instead of using the **<form>** and **<orth>** elements, not only in accordance with the editorial view (phrases are in italics while entry lemmas are in bold capitals), but also because of their specific status.

Conversion rules can still be used later to map these new elements to TEI DTD elements. Moreover, the conversion rules can be reversible (see section 3), although in practice, the most specific DTD is more likely to be used for input (making conversion from a proprietary DTD to the TEI DTD is more natural[10]). There are two main types of element creations: *creation from attribute-value pairs* and *element specialization*.

Creation from attribute-value pairs. In certain cases, the information represented in the TEI DTD using attribute-value pairs is transformed, in which case the attribute becomes a tag and the value becomes the textual content of the element. More precisely, two cases arise:
(1) The transformation is unconditionally performed, whatever content occurs within the element. For example, the homograph number will systematically be tagged as a **<hnum>** element, whatever the element content (the TEI encoding is on the left, the PLI on the right):

    <FORM hom=1>...</FORM>   ↔   <FORM><HNUM>1</HNUM>...</FORM>

(2) The transformation depends on the element content. For example, we distinguished[11] the sense recursion (a sense element is directly included within a sense element) from the sense specialization (a sense element is introduced as a specialization behind a definition). For example, we can see below s.v. IDÉAL that the sense introduced by the diamond sign is a sense refinement, and not a hierarchical marker.

> 1. IDÉAL, E, ALS [...] .1. Qui n'existe que dans la pensée et non dans le réel. *Monde idéal.* ◊ **Spécialt.** Qui relève de l'idée, qui est conçu par l'esprit [...]

Most sense specializations are introduced by diamonds or dashes and will be distinguished from plain hierarchical markers introduced by digits and letters. The two kinds of markers will be translated as follows:

    <SENSE NUM=X>...</SENSE>   ↔   <SENSE><SEP>X</SEP>...</SENSE>
    <SENSE NUM=Y>...</SENSE>   ↔   <SENSE><SENSENUM>Y</SENSENUM>...</SENSE>
            (where X is a dash or a diamond, Y is a letter or a digit)

From a structural standpoint, the two sense subdivisions are quite different.

Creation from element specialization. Here again, this kind of specialization is twofold depending or not on the element content :
(1) The transformation is systematic, whatever the element content. This kind of rules is the commonest (see list of conversion rules in Appendix A). For example, a specific tag <domain> was chosen for the PLI, since the typographic style for this class differs from the other usage marks (semi-bold small capitals instead of semi-bold small characters). The conversion rule is straightforward :

    <USG type="dom">*xyz*</USG>   ↔   <DOMAIN>*xyz*</DOMAIN>

(2) The transformation depends on the element content. For example, marks such as "par euphém(isme)" (e.g. s.v. QUELQUE PART), cannot be tagged as usage marks : they are included in the definitions, have a specific typographical presentation and are slightly different from other usage marks (they also indicate a semantic link with another sense in the entry).

QUELQUE PART adv. [...] . 3. Fam. Par euphémisme, pour désigner : a. les fesses ...

These marks cannot be blindly converted from the TEI DTD (since other stylistic usage marks do not appear with the same typographical properties) and the element content has to be checked :

    <DEF><LBL>*X*</LBL>...</DEF>   ↔   <DEF><SEMLBL>*X*</SEMLBL>...</DEF>
                (where $X \in$ "par euph.", ...)

In customizing the TEI DTD for the PLI, eighteen elements had to be added (See conversion rules in Appendix A). Only three, the ones pertaining to sense numbering and homographs, were created from attribute-value structures. The others are specializations, mainly needed for input ease and typographical rendition conformance to the original text.

5. Conclusion

The DTD proposed by the TEI for dictionary encoding is not the most suitable for an editorial view, which involves keyboard input and typographical recovery. Using the example of the *Petit Larousse Illustré*, we show that it is possible to enter and print dictionaries using specialized DTDs. If created in accordance with a few simple principles (stated in terms of DTD subsumption and transformations without information loss), the specialized DTDs allow for automatic conversion to the TEI DTD, which is still useful as an exchange format and allows for the utilization of generic tools. SgmlQL (Harié *et al.*, 1996) is an example of a language that can be employed to perform such a conversion.

6. Notes

[1]   Special thanks to Thierry Fontenelle who reviewed for us a previous version of this text.

[2]   The primary authors of the TEI dictionary chapter were Nancy Ide and Jean Véronis. The other members of the dictionary work group, who supplied many ideas and constructive criticism during our numerous discussions, were Robert Amsler, Susan Armstrong, Nicoletta Calzolari, Carol Van Ess-Dykema, John Fought, and W. Frank Tompa.

[3]   For instance, via a side effect, the ultra-specialized dictionary-specific tag <oref> becomes legal almost everywhere, including in the *header* whenever the dictionary sub-DTD is included (this problem was spotted in a discussion with Eric Peterson on the TEI-L list).

[4]   *Author/Editor* is a SoftQuad product, *GRIF* is a *GRIF,S.A.* product.

[5]   The authors acknowledge previous discussions on DTD subsumption with Nancy Ide. The formalization presented here is ours.

[6]   It should be noted that knowing whether or not a DTD subsumes another is a decidable problem, i.e. there exists an algorithm to answer this question (this results from McNaughton's 1967 theorems on parenthesis grammars).

[7]   We acknowledge fruitful discussions on PLI with students attending the computational lexicography class of the Diplôme Européen de Lexicographie at Lille 3.

[8]   *Author/Editor* in the present case. Note that some contextual typographical rules cannot be handled by *Author/Editor*, for example the fact that two brackets cannot co-occur in an entry for readibility reasons.

[9]   Thierry Fontenelle attracted our attention on the fact that many elements created for the PLI application would be useful for other dictionaries. For example, he noticed that elements like <orthintr> would be useful for Dutch or German dictionaries, while the <inflex> element could be profitably used in an English dictionary such as Cobuild where comparison degrees are systematically mentioned for adjectives (*great, greater, greatest*).

[10]   Conversion from the TEI DTD is less natural insofar as it requires that many attribute-value pairs be filled.

[11]   We thank Pierre Corbin for having attracted our attention on that.

7. References

Harié, S., Murisasco, E., Le Maitre, J., Véronis, J. (1996). SgmlQL: un langage de requêtes pour la manipulation de documents SGML, *Cahiers GUTenberg*, 24, 181-184. [see technical documentation at http://www.lpl.univ-aix.fr/projects/SgmlQL/].
Hausmann, F.J., Wiegand, H.E. (1989). Component parts and structures of general dictionaries: A survey. In Hausmann, F. J., Reichmann, O., Wiegand, H.E., Zgusta, L. (Eds.), *Wörterbücher: Ein internationales Handbuch zur Lexikographie*, Berlin: Walter de Gruyter, IV.36, 328-360.
Ide, N., Le Maitre, J., Véronis, J. (1993). Outline of a Model for Lexical Databases. *Information Processing and Management*, 29, 2, 159-186.
Ide, N., Véronis, J. (1995). Encoding dictionaries. In Ide, N., Véronis, J. (Eds.) (1995). The *Text Encoding Initiative: Background and Context*. Kluwer Academic Publishers, Dordrecht, 342 p.
ISO (1986). ISO 8879:1986. *Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)*. International Organisation for Standardization, Geneva.
McNaughton, R. (1967). Parenthesis grammars. *Journal of the Association for Computing Machinery*, 14:3, 490-500.

Sperberg-McQueen, C.M., Burnard, L. (1994), *Guidelines for Electronic Text Encoding and Interchange*, Text Encoding Initiative, Chicago and Oxford.

Sperberg-McQueen, C.M., Burnard, L. (1995). *TEILite : An introduction to Text Encoding for Interchange.* http://www-tei.uic.edu/orgs/TEI.intros/tei-5.tei.

The Danlex Group (1987), *Descriptive Tools for Electronic Processing of Dictionary Data,* Niemeyer, Tübingen, Lexicographica Series Maior.

APPENDIX A : New elements created for the *PLI* and conversion rules

Element	Definition	PLI Tags	TEI Tags	Comment
HNUM	Homograph Number	\<hnum>X\</hnum>	\< entry n=X> ...\</entry>	Treated in TEI by means of an attribute incompatible with editorial view.
SNUM	Hierarchical marker	\<sense> \<snum>X\</snum> \</sense>	\<sense num=X> / X contains digits or letters	Treated in TEI by means of an attribute (see above)
SEP	Sense specialization marker	\<sense> \<sep>X\</sep> \</sense>	\<sense num=X> / X contains a dash or a diamond	Treated in TEI by means of an attribute and not distinguished from hierarchical markers.
ORTH-INTR	"Orthographic introducer" after address (very common with phrases or pronominal verbs)	\<orth>X\</orth> \<orthintr>(Y)\</orthintr>	\<orth>X\</orth> \<orth type="intr">(Y) \</orth>	Nothing was proposed in the TEI DTD. Ex: IMMISCER (S')
INFLEX	Inflected form	\<inflex>Y\</inflex>	\<form type="infl" \<orth>Y\</orth> \</form>	TEI treatment (attribute on \<form>) seems tedious given high frequency of element. Ex:ICARIEN, ENNE
INFL-PRON	Pronunciation of the inflected form	(\<pron>X) \<inflpron>Y\</inflpron> (\</pron>)	\<pron type="infl"> Y\</pron>	Ex: IMPORTUN,E [ɛ̃pɔRtɔ̃, yn]
MOR-TYPE	Used for the inflectional type of the noun (generally 'inv.')	\<mortype>X\</mortype>	\<itype>X\</itype> / X ∈ "inv.", ...	Has different typography from inflectional type of verbs (for which \<itype> is used).
IRRPLUR PLUR	Irregular plural for nouns and/or multiword units	\<irrplur>\<plur>X\</plur> ...\</irrplur>	\<form type=infl> \<number>pl \</number> \<orth> X\</orth> \</form>	Has a specific typographic presentation and a specific distribution (after etymology).
PHRASE	Idioms or collocations	\<phrase>X\</phrase>	\<form type="phrase"> \<orth>X\</orth> \</form>	Very common in definitions.
LEXREL	Codified paradigmatic lexical relation	\<lexrel>X\</lexrel>	\<lbl>X\</lbl> / X ∈ "SYN","CONTR",...	
REFLEX	Reference pointed at (but not necessary defined in the same document)	\<reflex>X\</reflex>	\<xptr>X\</xptr>	
REFENTR	Reference to a key-word of the macro structure.	\<ref> \<refentr>X\</refentr> \</ref>	\<ref>\<ptr>X\</ptr> \</ref>	Ex: INCASIQUE adj. → *inca.*
SEMLBL	Semantic labels used to indicate a semantic relation between senses	\<semlbl>X\</semlbl>	\<lbl>X\</lbl> / X ∈ "pareuphém." ,"métonym." ...	
COM-MENT	Encyclopedic or usage comments	\<comment> X\</comment>	\<note type= "comment">X \</note>	Differs (typography and distribution) from \<note> and \<encycl>
DOMAIN	Domain mark	\<domain>X\</domain>	\<usg type="domain"> X\<usg>	Different presentation from other usage marks.
ENCYCL	Encyclopedic developments	\<encycl>X\</encycl>	\<note type="encycl"> ...\</note>	Can be quite long, contrary to \<comment>s).
REORTH	Orthographic form of related entry	\<re>\<reorth>X\</reorth> ...\</re>	\<re>\<form>\<orth>X \</orth>...\</re>	
HIROM	Highlighted forms in roman characters (between parts in italics)	\<hirom>X\<hirom>	\<hi rend="rom"> X\</emph>	

APPENDIX B : An example of tagged entry

INTRA-UTÉRIN, E adj. (pl. *intra-utérins, es*). Qui est situé ou qui a lieu à l'intérieur de l'utérus.

TEI version

```
<ENTRY>
    <FORM>
        <ORTH>INTRA-UTERIN</ORTH>
    </FORM>
    <FORM TYPE=infl>
        <ORTH>, E</ORTH>
    </FORM>
    <GRAMGRP>
        <POS>adj.</POS>
    </GRAMGRP>
    <FORM TYPE=infl>
        <NUMBER>pl</NUMBER>
            <ORTH>intra-utérins</ORTH>,
    </FORM>
    <FORM TYPE=infl>
        <NUMBER>pl</NUMBER>
        <GEN>fem</GEN>
        <ORTH NORM="intra-utérines">es </ORTH>
    </FORM>
    <SENSE>Qui est situé ou qui a lieu à
            l'intérieur de l'utérus.</SENSE>
</ENTRY>
```

PLI version

```
<ENTRY>
    <FORM>
        <ORTH>INTRA-UTERIN</ORTH>
        <INFLEX>, E</INFLEX>
    </FORM>
    <GRAMGRP>
        <POS>adj.</POS>
    </GRAMGRP>
    <IRRPLUR>
        <PLUR>intra-utérins</PLUR>,
        <PLUR NORM="intra-utérines" GEN=fem>es</PLUR>
    </IRRPLUR>
    <SENSE>Qui est situé ou qui a lieu à
                l'intérieur de l'utérus.</SENSE>
</ENTRY>
```

373
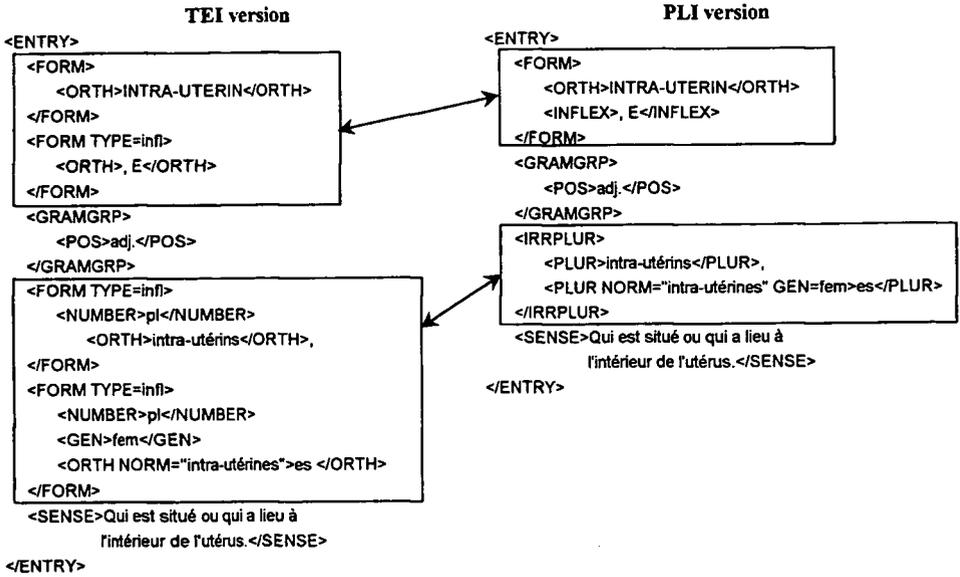
APPENDIX C : An example of SgmlQL rule

Example of one of the most complex rules (replacement of the TEI **<form>** element by the new **<irrplur>** element, including replacement of nested elements)

```
replace                                                    # replace every FORM by IRRPLUR in the entry
        every FORM as $form                                # (identifiers starting with $ are variables)
within
        $entry
by
(
        replace                                            # change ORTH in PLUR
                every ORTH as $orth
        within
        (
                remove                                     # remove NUMBER and GEN tags
                        every NUMBER, GEN
                within
                        element IRRPLUR content: content($form)    # this builds a new IRRPLUR element
        )
        by
                element PLUR
                        attr:
                        (
                          let $gen = text(first GEN within $form) in
                                        if $gen ne ""              # change GEN tag within FORM into
                                        then attr($orth) , (GEN=$gen)   # GEN attribute on PLUR
                                                        else attr($orth)
                        )
                        content: content($orth)            # content of PLUR same as content of ORTH
)
where
        $form->TYPE eq "INFL"                              # do all this only if FORM element is
        and not (empty (every NUMBER within $form))        # inflected and contains a NUMBER tag
```