Stefanie Heidecke, University of Bielefeld/ Bertelsmann Lexikon Verlag

SGML-Tools in the Dictionary-making Process – Experiences with a German-Polish/Polish-German Dictionary

Abstract

The paper gives a brief introduction to SGML and the motivation of its development. Different kinds of usage and three types of SGML applications are described: a) the analysis and annotation of existing texts, b) the standardization of existing texts, and c) the development of text structures. The annotational system is demonstrated by a short dictionary example. The consequences of SGML applications on the life-cycles of products (in our case a Polish-German/German-Polish dictionary (are described. The dictionary-making process has changed particularly with the use of SGML-aware tools, that allow authors, translators and editors to work directly within the SGML-documents. The experiences with this new technology and how it changes the everyday work in publishing houses are described.

1. SGML

First of all SGML (Standard Generalized Markup Language) was developed to meet the requirements of publishers to process their information in multiple formats and to disseminate them in a variety of media. It is called **generalized**, because it does not deal with the layout of every single peace of text, but with the function of the particular text in the context of the overall text structure. To define these functions in a document the text needs to be annotated or **marked up**. These annotations are basically easy to understand and handle. SGML does not deal with any fixed patterns, but offers the flexibility of a programming **language**. On the basis of this language annotational systems are developed – the so called DTDs (Document Type Definitions). This way SGML does not only provide the means to describe all types of information but also to consider the desired usage of them. When using SGML, a **standard** is guaranteed, because the definitorial language underlies the guidelines of the International Standard Organization (ISO 8879) since 1986.

Apart from this, the application of SGML is becoming more and more widespread due to the fact, that efficient SGML-tools, for e. g. converter, editors and typesetting systems, are developed.

2. An example

The following example demonstrates a simple structure description of a dictionary entry:

Haus *n* <-es; Häuser> house
Haus *n* <-es; Häuser> dom *m*

The SGML-annotation makes explicit what we implicitly know about the text structure of this article: 1) the text is an entry of a dictionary; 2) the first part is the head-word; 3) it belongs to the category noun, which needs to be specified by the gender; 4) additionally information about the declension are given, usually genitive singular and nominative plural; 5) there is no polysemy or partial equivalence, but a single equivalent; 6) the Polish translation requires to specify the gender. Concerning our SGML-element ENTRY it is necessary to distinguish between main or subsidiary entries, which is among others an important information for the layout. This kind of additional information is included in the **attributes** of an element.

```
<dictionary>
<entry  type = main> <head-word> Haus> </head-word>
   <n> <gender> &Genus-3 </gender>
      <decl> <gen-sg>-es</gen-sg>
         < nom-pl >H&a-uml;user</nom-pl> </decl>
      <single-n-descr>
         <list of transl>
            <tr-item>dom<gender> &gender-3 </gender><tr-item>
         </list of transl>
      <single-n-descr>
   </n>
</entry>
<entry> ... </entry>
...
</dictionary>
```

Fig. 1. SGML-document

Basically the DTD describes all possible entry structures. Hierarchically higher text elements are described with their subordinate parts until the level is reached, where text or named objects (**entities**) can be entered. The excerpt of the DTD below includes information about how often an element can or must occur (there must be at least one ENTRY element in a dictionary; there can be one or none declension information about a noun; elements of semantic discrimination are not compulsory but optional); the DTD defines the order of the elements and possible choices (see polysemy or single noun description within the element N), if attributes are necessary etc.:

```
<! element  dictionary      (entry +)>
<! element  entry           (headword, (%word-category;) )>
   <! attribut  entry       type (main | subsidiary)>
<! element  headword        (#pcdata)>
<! element  N               (gender, decl ?, (Npolysemy | single-Ndescr) )>
<! element  gender          (#pcdata)>
<! element  decl            (gen-sg, nom-pl)>
<! element  gen-sg          (#pcdata)>
<! element  nom-pl          (#pcdata)>
<! element  single-Ndescr   (explanation ?, Ntrans-list, example ?)>
<! element  Ntrans-list     (Ntrans-item +)>
<! element  Ntrans-item     (#pcdata, gender)>
<! element  explanation     (domain *, style *, synonyms *, collocator *)>

<! entity   word-category   "n | v | a | adv | p | nr | cj | art |  pro | ij  ">
```

Fig. 2 Extract of a dictionary DTD

The level of textanalysis, on which the DTD is based, depends on the intended applications. The purpose might be to guarantee an automatcally installed uniform layout, to support text retrieval or text comparison on a rather content orientated basis etc.

3. Usage of annotated texts

The advantage of an external, functionally orientated structure description is, that the annotated texts may easily be transferred in different formats and different media (print product, CD-ROM, on-line multimedia formats like HTML, international standardized exchange formats, individual formats that may be demanded by a database etc.). As content and form are stored separately, it is possible to maintain one single version of the basic data set. Changes and updates of information have to be implemented just once, not in every single product, that is derived from the basic data.

Supporting system independent data storage and portability does not only concern the annotation of the data, but also the basic character set. The realization of special characters is always system specific. To avoid difficulties during the text exchange they are represented by named objects (**entities**), which are stored in separate lists. These entities are comparable to macros, that contain strings of signs or files. Concerning a dictionary the set of entities is particularly convenient, when it comes to change the metalanguage, prepare another edition etc.

3. Three types of SGML applications

1. Type
The first type of SGML applications is relevant for publishing houses, when it comes to making existing texts electronically accessible. The goal is to exactly translate the texts into SGML annotated forms:

1) Analysis of the existing texts
2) DTD
3) Annotation of the existing data. For this purpose efficient converters can be used.

As an example of this may be named the TEI P3 (Text Encoding Initiative), which established a family of DTDs for historic texts and literature, that serves as the basis for administrative and private projects for electronic encoding of texts (e.g. 'Gutenberg Projekt').

2. Type
The second type starts out from existing texts but uses the process of the DTD development to control, correct, and generalize the text-structure. This is how the Bertelsmann Pocket Dictionary Polish-German/German-Polish has been produced:

1) Analysis of the existing texts
2) Decisions for standardization
3) DTD
4) Annotation of existing data and textproduction.

3. Type
The third application takes the development of a DTD as basis, which may then be filled with content.

1) Development of text structure
2) DTD
3) Production

These applications often make use of the HTML (Hypertext Markup Language) to which SGML-texts may easily be converted. It gives a standard for distributed hypertext- and multimedia documents. They serve e.g. as basis for the WorldWideWeb in the Internet. This is of particular interest for publishers in their role of *content providers* in the net, but also for On-line-services, public relations etc. in this medium.

4. The Dictionary-Making Process with SGML-Tools

Example: The Bertelsmann Pocket Dictionary Polish-German/German-Polish

4.1 The German-Polish Dictionary

The starting point of the production of the German-Polish dictionary was the data set of a German-English dictionary. They existed in a type-setting-format.

1. The first step consisted of analyzing the macro- and microstructures of the Bertelsmann Pocket Dictionary. As the development of DTDs demands a consequent structuring this step already revealed a certain non-uniformity in these data. That required editorial standardization decisions, e.g. standards concerning particularly pronouns and articles (listing of female and male forms, translation of declined forms or cross-references and so on); the order of elements to discriminate meanings (synonyms, typical object/subject) and additional features (register, region); uniform treatment of subcategorizations etc.

2.The second step was to convert the data into an SGML-annotated format. This could partly be done by an SGML-converter, but needed a manual "post-conversion", which has been carried out in a Windows-SGML-editor. This step again led to the adjustment of some details of the DTD.

The semi-automatic conversion basically consists of annotating the text with the correct SGML mark-up. This work can be done by some-one, who is a bit familiar with the dictionary structure and Windows editors. One does not need to be a SGML-specialist. The SGML-editors have the advantage, that they permanently control whether the structures built, agree with the DTDs or not. To construct incorrect structures is not possible! This minimizes the work of post-editing and proof-reading enormously.

3. Once the whole document is correctly annotated it is very easy to derive dictionaries with other target languages from these data:

1)The DTD has to be adjusted to the target language. As a Slavic language requires verbal aspect specifications, a corresponding element was included, as well as an element for the aspect partner(s).
2)The metalanguage can automatically be changed (alternating the affixes of elements, changing the attribute values, changing the entities).
3)The translator can fill in his translations directly into the SGML-document. In our case he merely had to replace the English by the Polish text. There is no need for an intermediate level between the author's work and a final proof-reading.

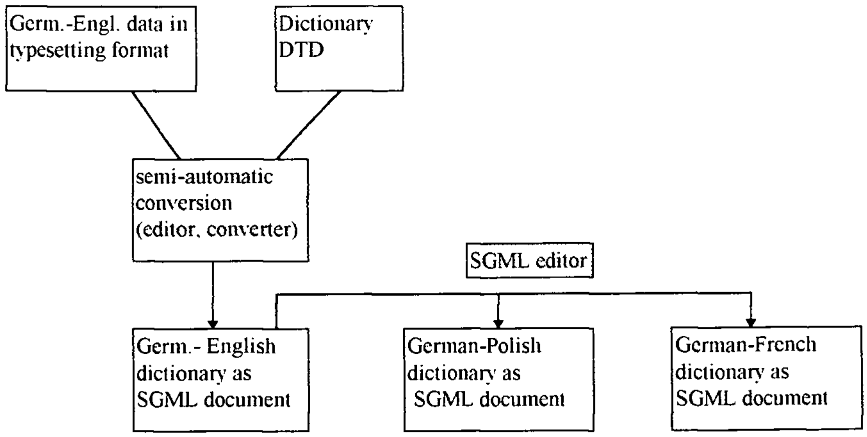
Germ.-Engl. data in typesetting format
Dictionary DTD
semi-automatic conversion (editor, converter)
SGML editor
Germ.- English dictionary as SGML document
German-Polish dictionary as SGML document
German-French dictionary as SGML document

Fig. 3 Dictionary-making process German-English/Polish/...

4. SGML-aware typesetting or DTP software or conversion tools guarantee a systematic realization of element and entity layouts. Once the automatic transformation process is correctly installed, the proof reader does not have to pay any attention to the element layout anymore and can concentrate fully on the content and orthography.

4.2. The Polish-German Dictionary

1. The dictionary with Polish as a source language demanded a lot of conceptional work on the general guidelines, since we had no data to

start with. The decision, about how each class of words should be treated, had to satisfy the language specific requirements and the philosophy of the dictionary series. This work is of course part of every dictionary making process, but we experienced that the rigorous structure descriptions of the DTD forced us to be highly explicit from the very beginning.

2. The testing period and the start of the author's work permanently led to refinements of the DTD. Until the final guidelines and DTD have been settled, one has to calculate about four months.

3. Particularly the author needs about 4–5 days of training and some consultations during his/her work. There is no knowledge of SGML needed, but building up the structures of articles is not quite trivial. According to our experiences so far an author needs about $1^1/_2$ as much time as if he/she would write or type a manuscript in the traditional way. It is very useful though, not to separate the author's work and recording of the data, because the SGML-editor offers a "guided" production process. As structure and content are highly dependent on each other – particularly in dictionaries – it makes no sense to separate these processes. Because of the permanent rules checking the author can control and correct his/her own work right away.

4. The translators' work consists in filling in the slots in the SGML-files.

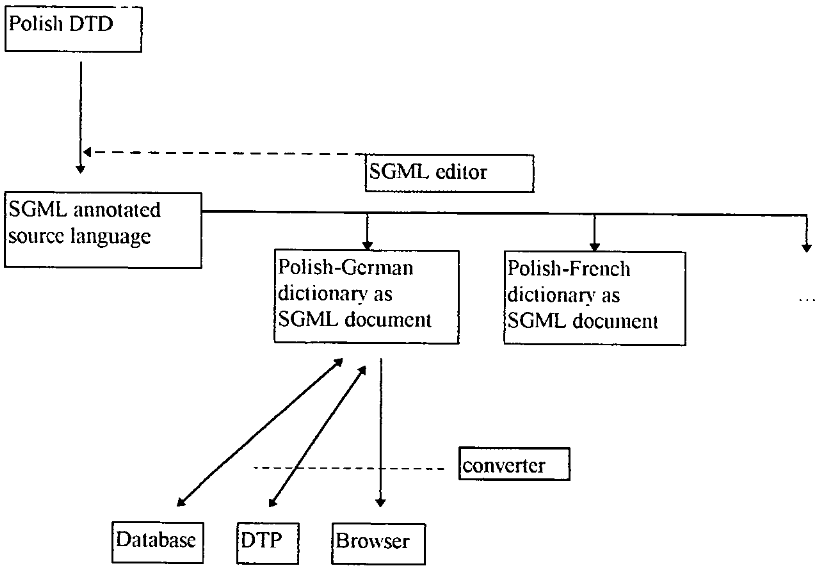
Polish DTD
SGML editor
SGML annotated source language
Polish-German dictionary as SGML document
Polish-French dictionary as SGML document
...
converter
Database
DTP
Browser

Fig. 4 From textproduction to system specific realization

5. Summary

5.1 Consequences for the editor's work

- The conceptual period until the final DTD has been developed and the actual work of the author can begin is longer than in the traditional dictionary making process.
- Working with the SGML-editors guarantees structural correctness in all evolutional states. That concerns the work of authors, translators, editors, layouters etc.
- Therefore the editors work on structural uniformity is minimized. This allows to concentrate more on the translation quality and orthography.
- The layout of the whole document and of each element is automatically installed.
- To use the SGML-editor particularly authors but also editors need some training. One has to think more in terms of function and structure than orientate oneself on the form. This reorientation proved to be a bit of a challenge, but a rewarding one!
- A tighter cooperation between typically EDP domains and typically editor domains is needed.

5.2. Consequences for the Publisher

- As the example demonstrated, it is easy to stock up to a broader language repertoire, once the basic data of one dictionary are annotated to SGML-texts.
- Shortening and extending versions is supported.
- The SGML-data can be converted automatically into any desired format.
- One substance can be transferred into different media realizations.
- The data storage is independent of the rapidly changing software technologies.
- Very efficient data-processing tools can be applied (SGML-aware editors, converters, DTP software).

References

Bryan, M. 1989. *SGML: author's guide to the Standard Generalized Markup Language.* Wokinham, England: Addison-Wesley.

Goldfarb, C. F. 1992. *The SGML handbook.* Oxford: Clarendon Press.

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