A Data Model and its Implementation for a Web-Based Language Learning System

Johann Gamper
Free University of Bozen-Bolzano
Dominikanerplatz 3
39100 Bozen, Italy
johann.gamper@unibz.it

Judith Knapp
European Academy of Bozen-Bolzano
Drususallee 1
39100 Bozen, Italy
judith.knapp@eurac.edu

ABSTRACT

At the European Academy of Bozen-Bolzano we are currently developing an adaptive Web-based language learning system for the German and Italian languages. In this paper we focus on the development and implementation of a data model for this system. While in the general context of e-learning the basic building blocks – often called learning objects – represent domain concepts, for our learning material the basic building blocks are words and expressions. Moreover, these pieces of data are highly interlinked. This situation requires a very fine-grained data model which stores meta-information at the word level and below. We present such a data model and discuss its implementation using XML.

Keywords

Computer-assisted language learning, data modelling, semi-structured data, XML

1. INTRODUCTION

It is widely accepted that learning can be supported efficiently by so-called new technologies. They offer exciting and powerful new features to present complex information, to communicate between students and teachers, to adapt the content to the individual learner, to access the learning material from anywhere and anytime, etc. Recent investigations have shown that learning with technology support might be motivating and efficient [4, 12, 15].

Language learning is a specific sector which definitely profits from new technologies. The WWW with its huge amount of freely accessible authentic language resources combined with technologies such as natural language processing, automated speech recognition, or adaptation is an ideal place to offer courses for autonomous language learning. In fact, more and more systems arise which combine the advantages of the Web with Artificial Intelligence technologies [7].

At the European Academy of Bolzano we are currently developing an adaptive Web-based language learning system called ELDIT. ELDIT offers individual language courses for autonomous learners by combining vocabulary acquisition with reading. It includes interactive exercises and a contact forum such that learners can collaborate with each others. Currently, the system is implemented for the German and Italian languages. A general overview of the system is described in [8].

In this paper we will focus on the development and implementation of a data model, which allows to store such complex, semi-structured information sets as required for our language learning system. Language learners need very fine-grained linguistic and semantic information about the target language. XML turned out to be an expressive language which allows an explicit description and encoding of complex information sets at the required level of detail. At the same time, the information has to be presented in an intuitive and clear way to the learner. New technologies including hypertext, multimedia, and adaptation techniques facilitate this knowledge transfer process by opening new doors for structuring and presenting learning material.

The paper is organized as follows. In section 2 we provide a short overview about ELDIT. In section 3 we analyse learner demands to a language learning system. Section 4 presents our detailed data model which allows to fulfill these learner demands. In section 5 we describe the current implementation of the data model and the data authoring process. Section 6 explains our motivation to choose XML as data representation language and discusses related work.

WWW2003, May 20–24, 2003, Budapest, Hungary.
ISBN 963-311-355-5.

2. THE ELDIT LANGUAGE LEARNING SYSTEM

The ELDIT language learning system for German and Italian has been conceptualized and partially implemented over the last three years. The initial idea was to develop a so-called learners' dictionary, which is a dictionary especially designed for language learners: the vocabulary coverage is limited, word definitions are simpler and often supported by a picture, carefully selected lexicographic patterns and examples show the typical use of a word, etc. Later on, other modules have been developed and integrated with the dictionary. The result is a full-fledged language learning system for intermediate and advanced learners, which has a strong focus on vocabulary acquisition and use. The overall system can be divided into the following modules:

- Dictionary
- Text corpus
- Exercises
- Tandem
- Lanugage tutor

2.1 Dictionary

The dictionary module is a union of several learners' dictionaries, currently a German and an Italian one. Each dictionary contains approximately 3000 word entries. Each word entry represents a huge amount of information – semantic information which helps

the learner to comprehend the right meaning of a word and syntactic information which helps to use the word correctly [2]. All these pieces of information are carefully selected and prepared up by language experts according to modern psycholinguistic criteria.

A dictionary entry is presented to the user in two frames (see figure 1). The left-hand frame shows the lemma of the word and a list of different word meanings, each of which is described by a definition, an example sentence, and an optional translation equivalent in the other language. The right-hand frame is organized in several tabs and shows additional, semantic and syntactic information such as word combinations, related words, linguistic difficulties, etc. The linguistic difficulties are also indicated by a kind of footnote numbers and shown in a small window on the place where they occur.

For the dictionary module we implemented a search engine especially targeted at supporting language learners who might have difficulties with the correct spelling of words: First, it is possible to restrict the search process to different items, e.g. the learner can search in the examples of the idiomatic expressions, in the compound words, in the definitions, etc. Second, the user can search a lemma or more complex expressions either directly or by some of its declined or conjugated forms, e.g. “chiesi” (asked) leads to the lemma “chiedere” (to ask). Third, problematic parts of a search expression might be omitted and replace by wildcards, e.g. “ca*are triste*a” matches with the collocation “cacciare la tristezza” (to banish sorrow). Finally, the search engine can detect spelling errors. For example, the word ”Schwiggermuiter” (“mother in law” written in a local German dialect) contains two spelling mistakes and is corrected to ”Schwiegermutter”.

From the lexicographic point of view, ELDIT is a completely new type of dictionary. On one hand, it is designed as two monolingual dictionaries in that each word meaning is described by a definition in the same language. This approach fulfills pedagogical demands which claim that it is better for the learner to remain in the target language. On the other hand, the definitions are extended with translation equivalents in the target language, a typical element of bilingual dictionary. This add-on fulfills learners’ demands, who usually prefer bilingual dictionaries. A second, innovative aspect stems from the combination of the two dictionaries. The translation equivalent serves as entry point to the corresponding part of the other dictionary. Note the German translations of the Italian word “casa” in figure 1, which are next to the Italian word definitions and are linked to the corresponding German word units.

2.2 Exercises

An important step towards a comprehensive language learning system is to extend the dictionary with simple gap-filling exercises. A gap-filling exercise is a text, where words have been removed and have to be entered by the learner. These exercises are used to apply and practice new vocabulary in the context of complete sentences.

We are reusing the rich set of information (definitions, the many example sentences, collocations, etc.) and generate these exercises automatically. For instance, an example sentence of a collocation can also be used as a gap-filling exercise, since we have explicitly encoded the occurrence of the pattern words within the example sentence. Multiple choice exercises can be created out of the information about word relation. Translation exercises can be provided since we have translation equivalents for all information pieces.

By reusing the search engine the system can provide automatic error correction and give meaningful feedback. Flexion mistakes can be found as well as spelling mistakes. The problem of synonymy of correct answers can at least partially be tackled with the help of the synonyms provided in the word fields. Remedial exercises can be provided by reusing the entire information in the dictionary.

2.3 Texts

ELDIT not only allows to practice language on small exercises, but provides a large amount of learning material to train language in a larger context and to produce target language output. For this purpose we have developed a text corpus consisting of 400 texts for each language which have originally been elaborated by the Goethe Institut of Milan as preparation material for the exams in bilingualism in South Tyrol. The texts are short articles selected from various magazines and books and contain approximately 150 words each. Every word is linked to the corresponding dictionary entry such that the learner can easily check unknown words – a very valuable feature in language learning [12, 15].

Each text contains a couple of questions which the learner has to answer with complete sentences in the target language. Note, that these exercises train the production of language and are different from the simple gap-filling exercises described previously.

2.4 Tandem

As the current version of ELDIT is not able to correct the student’s answers (except in the case of the gap-fill exercises) and to provide feedback, we are implementing an e-mail tandem which allows Italian and German native speakers to contact each others and to form learning partnerships: A notice board is added to the system on which learners can advertise their interest in a learning partnership. The learners can contact each others and correct the partner’s answers to the questions of the text samples.

Tandem learning is known as a powerful method in language learning, where the learning partners meet each other. Our e-mail tandem takes full advantage of the Internet and allows the communication between students independent of location and time. While oral communication is important and can only be trained in the traditional way, our tandem module is meant as an additional possibility to train written communication especially for those people who don’t have the possibility to meet a learning partner.

2.5 User Model and Tutor

The system is equipped with a user model which records all steps a user is carrying out during his or her learning sessions. The last module we want to include is an intelligent tutor which consults this user model, adapts content and links to the individual learners needs and preferences, and guides the learner individually through the learning material. Such adaptive systems have been proven to be an efficient and valuable learning tool, if the learner collaborates with the system [4]. Learning words (by reading the information in the dictionary and carrying out the corresponding exercises) and practicing texts (by reading the text and answering the questions) should occur alternately. Once a learner has learned some words, a suitable text is searched by the tutor on which these words can be applied. While the user is reading this text he or she might check some new unknown words. These words are the next ones the tutor will propose for study. Afterwards a new text is searched to apply these just practiced words. In this way it is possible to offer individualized, contextualized vocabulary acquisition, an approach which tries to fulfill pedagogical demands (“vocabulary acquisition by reading is less superficial”) and learner demands (“vocabulary acquisition by word lists is faster”).

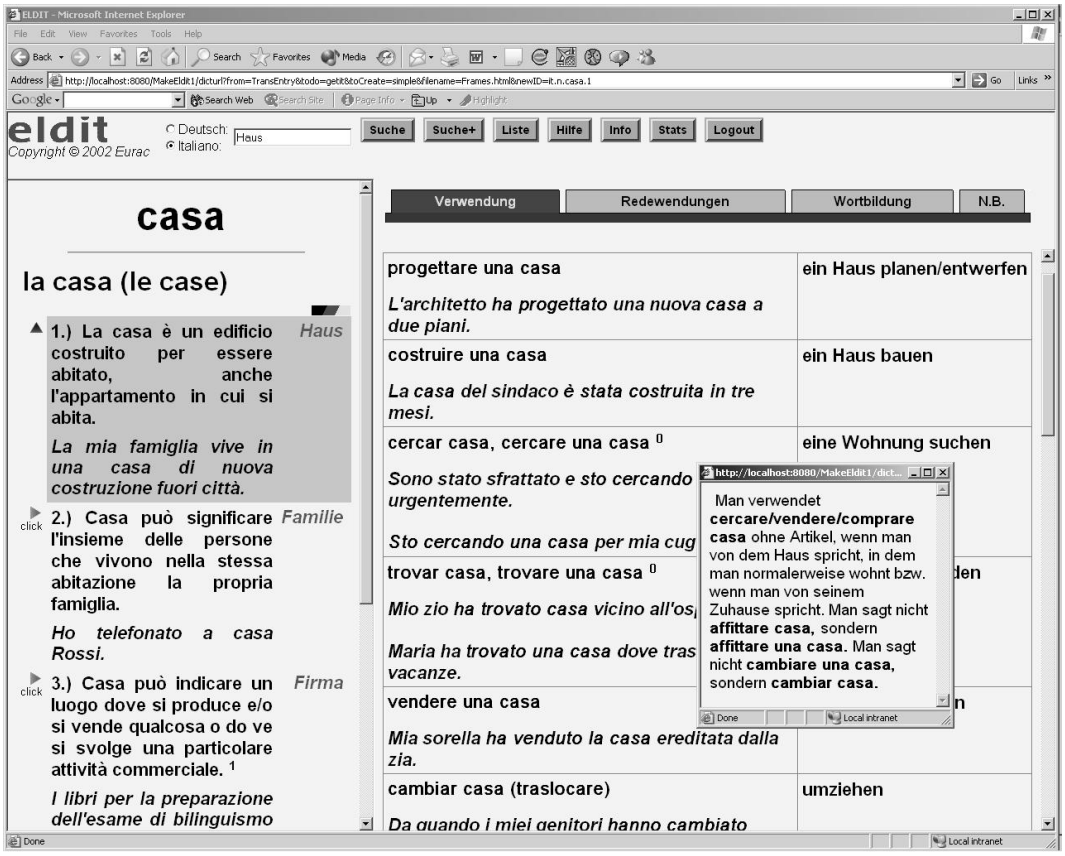

Figure 1: Screenshot of the dictionary entry for the Italian word ”casa”.

3. REQUIREMENTS FOR THE DATA MODEL

3.1 Learner Demands

One of the main design guidelines in the ELDIT project was to consider pedagogical and psycholinguistic learner demands which we classified into three groups [1]:

- *Support in decoding and encoding information:* Language learners need support in decoding and encoding information. Decoding information covers passive language skills and includes word and word meaning perception as well as understanding differences between synonyms. Encoding information covers active language skills and includes correct production of written and spoken language. In other words, encoding information means the correct combination of sequences of words to phrases according to grammatical rules of the language.
- *Intuitive presentation of information:* Natural languages are much more complicated than formal languages, and it requires a lot of information including typical patterns and sample sentences in order to explain sometimes very subtle differences between different items. These complex pieces of information have to be presented in an intuitive and learner-friendly way which supports the student as much as possible to create his/her mental models.
- *Personalization and individual guidance:* Given the huge amount of information stored in the ELDIT system, personalization and individual guidance might increase learning efficiency by pointing the learner to relevant parts of the huge learning space.

In order to respond to these learner demands we need to store rather detailed information about the language learning material and resources. In the following we discuss a few concrete examples which make these needs evident.

3.2 Examples

Word formation. An important aspect in vocabulary learning is to understand word formation, i.e. composition and derivation of words. Figure 2 shows four derivations of the German word “Haus” (house) as they are presented in ELDIT. In order to facilitate the comprehension of the derivations, we are highlighting the basis of a derivation (the part where the word is derived from), and the derivations are aligned in a specific way. Suppose a learner is reading a text and encounters the unknown word “Behausung”. If he/she is aware about the possibility and rules of word derivation, he or she might immediately understand that an encountered word, that contains the particle “haus” (such as “Behausung”) is a derivation of the word “Haus” and suspect that this word has something to do with a “Haus”, even if he or she does not know the exact meaning of the encountered word itself. Being able to provide this information requires a data model which explicitly distinguishes between prefixes, basis, and suffixes of a word.

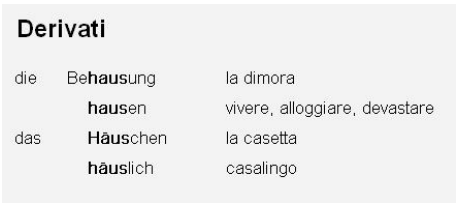

Figure 2: Derivations of the German word "Haus".

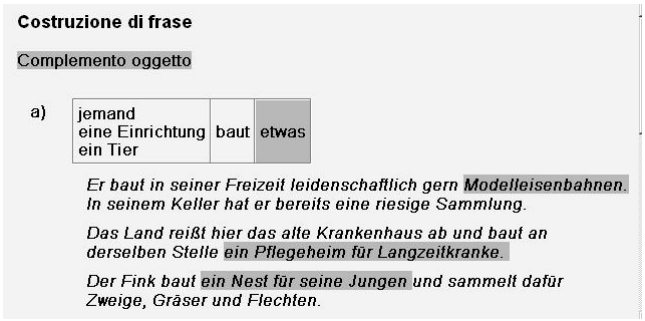

Figure 3: A typical pattern for the use of the verb "bauen".

Verb Valency. Another example is the problem of verb valency, which is about the correct use of a verb together with nouns and prepositions. The use of verbs can be described by a closed set of patterns, e.g. "qualcuno chiede qualcosa a qualcuno" (somebody is asking somebody something). We list these patterns in our dictionary and provide example sentences to illustrate them. Figure 3 shows a typical pattern of the German word "bauen" (build) together with three examples. In order to render the correspondence between the pattern and the example sentences more evident, the corresponding parts in the pattern and examples are highlighted, if the learner moves the cursor over the pattern or over one of the examples. In figure 3 the correspondence between the object "etwas" (something) in the pattern and the objects in all example sentences is made explicit.

Linking Text Pieces and Dictionary. The third example stresses the importance to establish links between the dictionary entries themselves and the words used in all kinds of text pieces, e.g. in the definitions, example sentences, or the text corpus. (see figure 4). A definition or example in the target language is useless if several words are unknown to the learner. Thus, we link all the words in the learning material to the corresponding dictionary entries such that the learner gets immediate access to the relevant information with a simple mouse click. For example, in figure 4 a learner has problems with the word "ufficio" (office) in the definition of the Italian word "edificio" (building). He/She clicked on the word "ufficio" (office) and gets in a separate window a short description of the unknown word composed of a definition, a sample sentence, and a translation. If the user wishes even more information about the new word, he/she can access the whole ELDIT entry via this new window. This type of links between text pieces and dictionary entries requires the encoding of information at the word level.

Content reuse. As a final example we want to mention the possibility to reuse elements for different purposes. This is possible, provided that enough meta-information is stored. For instance, an

Figure 4: Words in definitions are linked to the corresponding dictionary entry.

example sentence in a word combination can also be used to illustrate the meaning of some of the single words, and not just to illustrate the word combination itself. Moreover, all kinds of text pieces, namely definitions, example sentences, etc., can be used to generate gap-filling or multiple choice exercises.

4. ELDIT DATA MODEL

The previous section made it clear that a very fine-grained data model is necessary to be able to provide an appropriate support for language learners. In this section we give a detailed description of the ELDIT data model.

4.1 Characteristics of the Data

The main characteristics of the learning material and resources in the ELDIT system can be summarized as follows:

- *Semi-structured data:* Semi-structured data are characterized by the lack of a regular structure and clear schemata. The ELDIT data are typical examples of semi-structured data. There are text units of different size, lists of words, pictures, and sound files.
- *Detailed level of granularity:* While in traditional learning material the typical level of granularity is that of a learning object which represents a concept in the specific domain, language learning requires a more detailed level of data representation down to the level of words and part of words.
- *Highly interlinked elements:* The ELDIT data are further characterized by large number of links between the different pieces of information.

4.2 Data Model for the Dictionary

Figure 5 shows a simplified version of the data model of the ELDIT dictionary, using an extended entity-relationship notation with generalisation/specialization. The main entities in our domain of discourse are

- words,
- word senses, and
- word groups.

Each of these elements is composed of and related to a number of other entities. All entities are chunks of text which contain additional information as will be discussed later. Note that there is a clear distinction between the mainly lexical information about a word (represented by the word entity) and the mainly semantic information about the different meanings of a word (represented by the word sense entity).

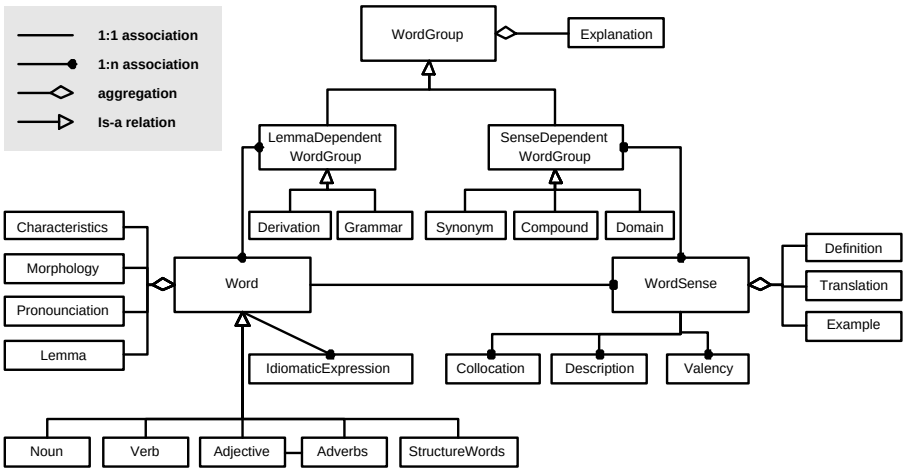

Figure 5: Simplified representation of the data model of the ELDIT dictionary.

Words are classified into different categories: nouns, verbs, adjectives, adverbs, and structure words. Each word entity is composed of various pieces of lexical information which is independent from a particular meaning of the word. This includes the lemma of a word (e.g. ”casa”), morphological information such as article and plural form (e.g. ”la casa, le case”), an adverb form in the case of adjectives, idiomatic expressions (e.g. ”a casa mia”), and optionally some remarks about special characteristics of the word such as linguistic difficulties and pitfalls.

Each lexical word form might have several meanings which we call *word senses*. A word sense entity is composed of a short definition in the same language as the word itself, one or more example sentences, and one or more translation equivalents in the other language. For example, the screenshot in figure 1 lists 3 different meanings of the Italian word “casa” with three different German translations. A word sense is further associated with additional information which helps to get a more comprehensive understanding of the word meaning as well as to use the word in the correct way. The additional information includes collocations, which are typical word combinations (e.g. ”progettare una casa”), a description element, which is a list of adjectives that typically occur with a word, verb valency, which describes the use of a verb together with subjects, objects, or prepositions, a prototypical picture, etc.

The third type of primary entity are groups of related words which we call *word groups*. According to new psycholinguistic studies in language learning, students are memorizing words in multidimensional networks of related words [3, 11]. Hence, we put a strong focus on clustering the information according to these psycholinguistic criteria and to show the words in relation to other words. In ELDIT these relations are not only named but also annotated, e.g. with explanatory information. For example, even if synonyms have the same meaning (by definition), in reality there are usually subtle differences, e.g. different synonyms are predominantly used in a different context. As another example we refer to the group of continents, where we not only list the names of the continents but also some additional information such as how persons living in a continent or how things belonging to a continent are called, e.g. “Afrika”, “Afrikaner”, “afrikanisch”. Such differences are important for language learning and shall be explained explicitly.

In our data model a first distinction can be made between word-dependent groups, which are independent of a particular meaning, and sense-dependent groups.

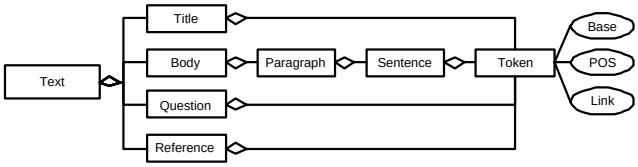

Figure 6: Data model of text units.

A typical example of a word-dependent group are derivations. A derivation entity groups all words together which are derived from the same base word. An example of a group of derivations is shown in figure 2. Another type of lemma-dependent word groups are formed by grammar units which mainly contain information about structure words such as pronouns, prepositions, or articles as well as grammatical aspects such as conjunctions, etc. Grammar units also include thematically related words, for instance the continents, the days of the week, etc.

Sense-dependent word groups cluster words depending on specific word meanings. Figure 5 shows just a few of them. Synonyms are groups of different words with the same meaning. Similar groupings of words in ELDIT are hyperonyms/hyponyms (more general/specific words) and antonyms (words with opposite meanings such as “cold” and “warm”). In linguistics such groups are called word fields. In the group of the compound words all words are grouped together which are composed of the same base word. The domain group tries to group words according to different domains, for instance sports, music, traveling, etc.

4.3 Data Model for the Texts

Short texts are an important resource for language learning, since they show the use of words in a larger context. Figure 6 shows the data model of the texts in ELDIT. Each text consists of a title, a text body, a reference element, and a couple of questions to be answered by the learner. The body is further divided into paragraphs, sentences and quotes (spoken language). In order to be consistent with the original texts, which are currently only available in paper form, some hints in brackets are included which provide translations to very difficult words that are ocuring within the text.

4.4 The Need for More Details

Figure 5 and 6 show the data models at a rather coarse level. To be able to provide the previously mentioned support, we need much more details.

4.4.1 Word-Level Annotation

Many of the main entities in figure 5 are composed of several sub-entities. For instance, idiomatic expressions and collocations are described by a pattern, some examples, and one or more translations. Verb valency is described by a pattern, some comments, and some examples. Derivations and compound words are described by a formation rule, comments and one or more translations. Synonyms, hyperonyms, antonyms, etc. are described by their names, relations, and differences, etc.

Most of the sub-entities consist of one or several short text sentences or some words. In order to reuse this information for different purposes and to evidence particular aspects of a language we have to annotate these texts at the word level and even below: We encode each word separately and tag all words with part-of-speech and lemma. This information can be used to partially automate the creation of links between words in the texts and the corresponding dictionary entries, and for the automatic generation of exercises from the text samples. The derivations and compound words are split up into prefix, basis, and suffix, which allows to make the different parts of a word evident to the learner (see figure 2) and to link e.g. prefix and suffix to the corresponding explanations about word formation. In some cases we add also style information to the words, e.g. in the example sentences of collocations we emphasize the words which belong to the pattern of the collocation.

4.4.2 The Link Structure

A particular characteristic of the ELDIT learning material is that all the pieces of information are highly interlinked. In particular, all words in definitions, example sentences, texts, etc. are linked to the corresponding dictionary entry in order to provide a fast access to meaning explanatory information. The part-of-speech information and the lemma added to each word help us to partially automate the generation of these links. Once the links are established, all information recorded for this dictionary entry can be provided on one single mouse click, e.g. information concerning the specific meaning of a word, morphological information (e.g. about number and case), or other illustrative or explanatory hints.

5. IMPLEMENTATION USING XML

For the implementation of the data model we were seeking for an expressive language which fits our needs to represent semi-structured data, but at the same time is simple to use and robust to frequent changes. We decided to use XML as uniform data representation in ELDIT for reasons that will be explained in section 6.

5.1 Document Type Definitions

Several mechanisms have been developed to specify structure and data types in XML. So-called document type definitions (DTD) are an integral part of the first XML-1.0 standard and provide a simple way to specify the structure of XML data. Since most of our data are words and text pieces which can be represented as strings, we decided to use DTDs instead of more powerful, yet more complicated mechanisms such as XML Schema, which distinguishes between several pre-defined data types and allows to model user-defined data types and inheritance hierarchies.

It is rather straightforward to map the data model described above into document type definitions. The main entities such as the different types of word units (nouns, verbs, adjectives, etc.) and the texts are represented by specific DTDs. The other entities are represented by XML elements in these DTDs. The properties of the entities are represented by XML attributes. To all elements we assigned a unique ID attribute, which in combination with a reference attribute is used to represent the links between words and dictionary entries.

Figure 7 shows a part of a DTD instance of an ELDIT text. Each word is annotated with its base form, part-of-speech, and an ID which refers to a dictionary entry.

5.2 The DXML Package

There are many possibilities to handle XML-documents. XSLT is a rule-based language, which allows to specify a set of rules for the translation of XML files into another format including HTML. XT and Cocoon are two well known XSLT translators. SAX is an event-driven interface to access XML documents, which has the advantage that large documents can be parsed and new data structures can be constructed. Tree-based APIs such as DOM, JDOM, or DXML allow reading and writing XML documents as if they were regular Java objects. This approach is the most flexible one, but puts great constraints on system resources if the XML document is very large, as the entire document is loaded into the main memory. JDOM is able to read only parts of the document, hence combines the advantages of an event-driven and a tree-based API.

Since the overall size of a dictionary entry, a word group or a text is not very large (5 to 20 KB), we decided to use the most flexible approach, namely a tree-based API. DXML [13] is a tree-based API available as Java package that allows reading and writing XML files as if they were regular Java objects. The first step in using DXML is to run the `xgen` utility on a specific DTD. This generates a Java package which contains an interface and an implementation class for every element defined in the DTD. The name of the package is the same as the name of the DTD file. The name of each interface and class derives from the name of the corresponding element. The interface provides methods to retrieve the attributes of the element and to access all its child elements in different ways. Unlike with DOM or JDOM, no type conversion is necessary, which yields a clean programming code.

The DXML package further provides classes and methods to create a new XML document, to open an existing XML document, to traverse an XML document, and to save its content. If a document does not match the DTD, a clear error message is provided which includes the exact location of the error.

5.3 Searching with Lucene

In order to search the XML files we use Jakarta Lucene, a set of Java APIs that provides the possibility to index and search text documents [6]. In order to provide structured search possibilities in the dictionary (which means that the search operation can be limited to specific parts of a word entry, i.e. one single XML element) each single XML element has been inserted separately as a document into the index and can be referenced by its ID. Moreover, the text is not only inserted in its natural, inflected form, but also in its base form, i.e. not the words itself, but the lemmatized form recorded in the base attribute of an XML-encoded word is inserted into the index, which allows to match conjugated or declined search expressions via their base form.

5.4 Authoring Learning Material

Data acquisition is generally known as a very time-consuming process. ELDIT is no exception, in particular due to the many pieces of information which are highly interlinked. Therefore, we were seeking for a methodology to reduce the burden of authoring

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE text (View Source for full doctype...)>
- <text level="ab" freq="0" domain="general" id="de.ab.general.005" index="005" lang="deutsch">
  - <title id="de.ab.general.005.title">
      <w idref="de.g.praeposition.1" base="in" id="de.ab.general.005.title.w000" ctag="APPR">In</w>
      <w idref="de.t.stadt.1" base="Amsterdam" id="de.ab.general.005.title.w001" ctag="NE">Amsterdam</w>
      <w idref="de.v.sein.1.lemma" base="sein" id="de.ab.general.005.title.w002" ctag="VAFIN">ist</w>
      <w idref="de.g.artikel.1" base="d" id="de.ab.general.005.title.w003" ctag="ART">der</w>
      <w idref="de.n.rad.1.sense2.comp0.pbs0" base="Radfahrer" id="de.ab.general.005.title.w004" ctag="NN">Radfahrer</w>
      <w idref="de.n.könig.1.lemma" base="König" id="de.ab.general.005.title.w005" ctag="NE">König</w>
    </title>
  - <body id="de.ab.general.005.body">
    + <p id="de.ab.general.005.body.p000">
    - <p id="de.ab.general.005.body.p001">
      + <s id="de.ab.general.005.body.p001.s000">
      + <s id="de.ab.general.005.body.p001.s001">
      + <s id="de.ab.general.005.body.p001.s002">
      + <s id="de.ab.general.005.body.p001.s003">
      - <s id="de.ab.general.005.body.p001.s004">
        - <q id="de.ab.general.005.body.p001.s004.q000">
            <w idref="KKK" base="Hey" id="de.ab.general.005.body.p001.s004.q000.w000" ctag="NN">Hey</w>
            <w idref="SSS" base="," id="de.ab.general.005.body.p001.s004.q000.w001" ctag="$,">,</w>
            <w idref="de.g.artikel.1" base="d" id="de.ab.general.005.body.p001.s004.q000.w002" ctag="PDS">das</w>
            <w idref="de.v.sein.1.lemma" base="sein" id="de.ab.general.005.body.p001.s004.q000.w003" ctag="VAFIN">ist</w>
            <w idref="de.g.possesivpronomen.1" base="mein" id="de.ab.general.005.body.p001.s004.q000.w004"
              ctag="PPOSAT">mein</w>
            <w idref="de.n.fahrrad.1.lemma" base="Fahrrad" id="de.ab.general.005.body.p001.s004.q000.w005"
              ctag="NN">Fahrrad</w>
            <w idref="SSS" base="!" id="de.ab.general.005.body.p001.s004.q000.w006" ctag="$.">!</w>
          </q>
        </s>
      - <s id="de.ab.general.005.body.p001.s005">
```

Figure 7: Part of a text document encoded in XML.

learning material as much as possible. The authoring process from the raw data created by the authors to indexed Java files which can directly be accessed by the ELDIT system is a four-step process:

1. Creation of the raw XML file
2. Validation and transformation of the raw XML file
3. Creation of a Java object
4. Creation of the index

The first step is that the linguists insert the raw data with an XML editor. We are currently using the freeware EXml developed by CUESoft, but any other XML editor could be used as well. Even with a sophisticated editor, the manual creation of these highly structured XML files would be too time-consuming. To reduce this burden, we defined simplified DTDs for the manual editing process (see figure 8a): First, special characters are used to separate fine grained information pieces, e.g. an underscore separates the prefix, basis, and suffix of a derivation. Secondly, data which can be derived automatically are not entered by hand and, hence, the corresponding elements are deleted from the DTD. The EXml editor with the simplified DTDs has been appreciated by the linguists as a useful editing environment.

The second step is to validate the raw XML files created by the linguists and to translate them into the original form, where all pieces of information are explicitly encoded in XML (see figure 8b). The translation process performs the following steps. First, a unique ID is added to each single element. Second, the special characters are removed and the corresponding pieces of information are explicitly encoded as XML elements. Third, the single words appearing in the definitions, translations, etc. are lemmatized and POS-tagged. Fourth, the words are linked to the corresponding dictionary entries, i.e. the href-attributes are inserted.

The third step in the data acquisition process is to compile the XML files into Java objects and to store them as Java OS files using

```
<derivation>
  <pattern>
    <content>die Be_haus_ung</content>
  </pattern>
  <translation>
    <content>la dimora</content>
  </translation>
</derivation>
```

(a)

```
<derivation id="de.n.haus.1.deriv2">
  <pattern id="de.n.haus.1.deriv2.patt0">
    <article>die</article>
    <praefix>Be</praefix>
    <basis>haus</basis>
    <suffix>ung</suffix>
  </pattern>
  <translation id="de.n.haus.1.deriv2.trans0">
    <w  type="content" base="il" ctag="S"
        href="it.g.articoli.1">la</w>
    <w  type="content" base="dimora" ctag="N"
        href="it.n.dimora.1.lemma">dimora</w>
  </translation>
</derivation>
```

(b)

Figure 8: Parts of XML files for the dictionary entries: (a) raw version created by the linguists and (b) the final version which is created automatically.

object serialization. This step is only required to gain efficiency, since parsing XML documents is rather time consuming even for small documents. ELDIT can now access a dictionary entry by reading the corresponding Java object which represents the XML file and provides methods to retrieve the single elements.

The last step is to create an index which supports a detailed search in the dictionary.

6. DISCUSSION

6.1 Why XML?

XML has several attractive features as a data representation and communication language, especially for Web applications. The most important aspect for ELDIT is the *flexibility* of XML, which allows an easy representation of semi-structured, highly interlinked data at a very fine-grained level of detail. The ELDIT data are good examples of such complex data sets which lack a regular structure. For instance, we have lists of words, smaller and larger text pieces, pictures, sound files, etc., all of which are connected at the word level. Another aspect of flexibility concerns the fact that a DTD can easily be modified or extended if the requirements change during system development. This is quite usual for prototyping and happened several times in the development of ELDIT. Frequent modifications in a relational database would be more cumbersome and might easily lead to redundancies.

Although it is very flexible, XML is *simple, human readable, and easy to use*. These advantages over other data representation languages are again very important in the prototyping phase when the requirements often change and error debugging has to be done. Moreover, we experienced that the communication with the linguists has been facilitated by the fact that the XML data model was easy to understand, since the data are represented in a natural way leaving coherent information together. This is a quite important aspect, since knowledge engineering is known as a difficult task, where the knowledge engineer has to mediate between domain experts and the formal representation of the domain in a computer. Thereby, a good knowledge representation language facilitates the communication between the knowledge engineers and the domain experts.

Another advantage of XML is the *open standard and the free availability* of an increasing number of tools for processing XML data. This brings much flexibility in the implementation process. System developers can test different tools and choose the most suitable ones to implement a running system at low costs. This helps not only to reduce the development costs, but to a certain extent allows adapting the system to individual needs. For example, each ELDIT author can use its favorite XML editor to enter the data.

Finally, XML follows *a strict separation between the structure and the presentation of the data*. While XML specifies only the structure of the data, other standards have been developed in parallel which facilitate the presentation of XML data in Web applications, e.g. XSLT and CSS. Thanks to the XSLT language it is straightforward to transform XML data into any other data format including HTML which is still the main language to render information on the Web. As XML is likely to become the future standard for data representation on the Web, the transformation into HTML might become obsolete and our data can directly be rendered by the client application.

6.2 Related Work

In the last couple of years a lot of research has been done to develop systems, data models, and standards for Web-based learning in general, e.g. WebCT, Hyperwave, KBS Hyperbook, InterBook, SCORM, LOM, etc. Several authors developed data models which are especially designed to represent and share teaching material over the Web. Henze et al. [9] describe a data model to support constructivist learning in the KBS Hyperbook system. Süß et al. [17] describe a meta-modeling approach to adaptive hypermedia-based electronic teachware that focuses on document structures and navigational services. The main difference of these systems to our approach is the level of granularity. While in general the basic building blocks are learning objects which represent a domain concept, we need a more fine-grained model which breaks the learning material down to the level of single words and further.

Recently, the advantages of XML and related standards have also been exploited in other language learning systems. The KirrKirr system is a Web-based application which allows users to explore a Warlpiri (a Central Australian language) dictionary [10]. The data are represented in XML. XSL is used for enhanced customization, and the XQL language is used to query the dictionary entries.

CoCoaJ is a system for writing in Japanese [14], which allows students and teachers to exchange documents over the Internet. The system includes a writing error analysis model, through which typical morphological errors can be detected. For the annotation of documents with remarks and comments, the eXtensible Communicative Correction Mark-up language (XCCML), which is based on XML, has been developed.

The authoring tool WURLE [18] and the multi-agent learning system IDLE [16] are two learning tools which use a similar approach for data management as we do in ELDIT. Educational content is provided in chunks of XML data, which are automatically linked according to an indicated lesson plan represented in a dependency graph. Both systems include a user model and adapt content presentation to each individual learner. In ELDIT, however, we have encoded the data in much more detail, namely down to the level of single words. While this allows to provide very specific information to the learner, we had to put a strong focus on supporting the authors in the content creation process. Thus, we use simplified DTDs for entering data by the user, which in a second step are extended with automatically derived information and transformed into the final form. Moreover, the fine-grained annotation allows us to reuse data and to provide a better adaptation to the individual learner.

The main difference between our dictionary and commonly known dictionaries or lexical databases, such as WordNet [5], lies in the different objectives of the systems. As a consequence the stored information is partially different, but in particular the elaboration and presentation of the material has to be different. Dictionaries and lexical databases are usually intended as a reference tool and try to describe knowledge as completely as possible. The ELDIT dictionary is a so-called learners dictionary, especially designed for language learners. Therefore, it includes only the basic vocabulary for each language and only the most important usage patterns. Special attention is given to provide a lot of well-designed illustrative material, such as examples, pictures and sound files. In order to fulfill pedagogical demands and to support the learner as much as possible, the data pieces have been encoded at a very low level of granularity. Lexical databases such as WordNet provide valuable input for the creation of the ELDIT learning material, but cannot be used directly for didactic purposes.

7. CONCLUSION

In this paper we presented a data model and some implementation issues for ELDIT, a Web-based language learning system for the German and Italian languages. Our learning material shows some important characteristics which differ from traditional sys-

tems: the data are semi-structured and highly interlinked and have to be annotated at a very fine-grained level of detail. In fact, we have to encode information at the level of single words and even below. This level of detail is needed in order to support the language learner as much as possible, and at the same time it allows to reuse the learning material for several purposes. While the current version of ELDIT covers only the two languages German and Italian, the data model is general enough to include additional languages.

Adopting a rapid prototyping approach, we were seeking for a simple, yet expressive language to implement our data model, which at the same time is robust to frequent changes and facilitates the knowledge engineering process. XML shares these properties and turned out to be a good choice for the implementation. While efficiency problems often require the use of a powerful database system, the representation of the XML files as Java objects is currently efficient enough for the ELDIT system.

A preliminary version of the ELDIT system is accessible at http://www.eurac.edu/Eldit. The dictionary is almost complete. Other modules such as the text corpus, the grammar sections, and the word groups are under development and will be online in the end of 2003.

8. REFERENCES

[1] Andrea Abel, Johann Gamper, Judith Knapp, and Vanessa Weber. New answers to old questions about lexicon acquisition and dictionary use. In *Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2002)*, 2003. Accepted for publication.

[2] Andrea Abel and Vanessa Weber. ELDIT, prototype of an innovative dictionary. In Ulrich Heid, Stefan Evert, Egbert Lehmann, and Christian Rohrer, editors, *Proceedings of the 9th EURALEX International Congress on Lexicography (EURALEX'00)*, Stuttgard, Germany, 2000.

[3] Jean Aitchison. *Words in the Mind: An Introduction to the Mental Lexicon*. Blackwell Publishers Ltd, Oxford, UK, second edition, 1994.

[4] John Eklund and Peter Brusilovsky. The value of adaptivity in hypermedia learning environments: A short review of empirical evidence. In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia held in conjunction with Hypertext´98*, 1998.

[5] Christiane Fellbaum, editor. *WordNet — An Electronic Lexical Database*. MIT Press, 1998.

[6] The Apache Software Foundation. Jakarta Lucene. http://jakarta.apache.org/lucene/docs/index.html, 2002.

[7] Johann Gamper and Judith Knapp. A review of intelligent CALL systems. *Computer Assisted Language Learning*, 15(4):329–342, oct 2002.

[8] Johann Gamper and Judith Knapp. A Web-based language learning system. In *Proceedings of the 1st International Conference on Web-based Learning (ICWL2002)*, pages 106–118. Springer, Lecture Notes in Computer Science, 2002.

[9] Nicola Henze and Wolfgang Nejdl. Adaptivity in the KBS hyperbook system. In *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, 1999. Available from http://www.kbs.uni-hannover.de/~henze/paperadaptivity/Henze.html.

[10] Kevin Jansz, Wee Jim Sng, Nitin Indurkhya, and Christopher Manning. Using XSL and XQL for efficient, customised access to dictionary information. In *Proceedings of the Sixth Australian World Wide Web Conference (AusWeb2k)*, pages 167–181, Cairns, Australia, 2000. Available from http://ausweb.scu.edu.au/aw2k/papers/jansz/paper.html.

[11] Bernd Kielhöfer. Psycholinguistische Grundlagen der Wortschatzarbeit. *Babylonia*, 1996.

[12] Batia Laufer. Electronic dictionaries and incidental vocabulary acquisition: Does technology make a difference? In Ulrich Heid, Stefan Evert, Egbert Lehmann, and Christian Rohrer, editors, *Proceedings of the 9th EURALEX International Congress on Lexicography (EURALEX'00)*, pages 849–854, 2000.

[13] Inc. ObjectSpace. DXML. http://www.objectspace.com/products/prodDXML.asp, 2000.

[14] Hiroaki Ogata, Yoshiaki Hada, and Yoneo Yano. Computer supported online correction for collaborative writing. In *Proceedings of IS'00*, pages 576–583, Aizu-Wakamatsu City, Fukushima, Japan, 2000.

[15] Isabelle De Ridder. Are we conditioned to follow links? Highlights in CALL materials and their impacts on the reading process. *Computer Assisted Language Learning*, 13(2):183–195, April 2000.

[16] Yi Shang, Hongchi Shi, and Su-Shing Chen. An Intelligent Distributed Environment for Active Learning. In *Proceedings of the 10th International World Wide Web Conference (WWW10)*, 2001. Available from: http://www10.org/cdrom/papers/frame.html.

[17] Christian Süß, Burkhard Freitag, and Peter Brössler. Meta-modeling for web-based teachware managment. In *Advances in Conceptual modeling. Workshop on the World-Wide Web and Conceptual Modeling (ER'99)*, volume 1727 of *LNCS*, Berlin, 1999. Springer-Verlag.

[18] T.J.Brailsford, C.D.Steward, M.R.Zakaria, and A.Moore. Autonavigation, Links and Narrative in an Adaptive Web-Based Integrated learning Environment. In *Proceedings of the 11th International World Wide Web Conference (WWW2002)*, 2002. Available from: http://www2002.org/CDROM/alternate/738/.