Outline of a Database Model for Electronic Dictionaries

Nancy M. IDE [1], Jean VERONIS [1,2], Jacques LE MAITRE [2]

[1] Department of Computer Science
VASSAR COLLEGE
Poughkeepsie, New York 12601 (U.S.A.)

[2] Groupe Représentation et Traitement des Connaissances
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
31, Ch. Joseph Aiguier
13402 Marseille Cedex 09 (France)

Abstract

The growing avaibility of dictionaries in electronic form calls for a model sophisticated enough to represent the richness of entries and enable complex information retrieval. Electronic dictionaries are a special kind of object, intermediary between a text and a database. Textual models are not powerful enough to handle complex information retrieval, and conventional database models are not flexible enough to handle the richness of their information. In this paper, we outline a scheme for representing electronic dictionaries which departs from previously proposed models. In particular, it allows for a full representation of sense nesting and defines an inheritance mechanism which enables the elimination of redundant information. The model provides flexibility which seems able to handle the varying structures of different monolingual dictionaries.

1. Introduction

Dictionaries are not used as most texts are--that is, the information in a dictionary is not accessed linearly, from start to end. Instead, the reader searches for a given entry with a more or less binary strategy on the basis of a *key* (word), and retrieves one or more associated *attributes* (pronunciation, grammatical information, meaning, etymology, etc.). Such operations are much more like those typically performed on a database. Because of this similarity, there is considerable interest in representing dictionaries in an electronic database form. An electronic dictionary database not only facilitates retrieval, but also enables operations beyond those possible for a dictionary in printed form (for example, extract all definitions in which a given word appears). Therefore, publishers are interested in producing electronic counterparts of existing printed dictionaries for distribution to the public at large, via media such as CD-ROM. However, both publishers and the research community are interested in creating even more sophisticated databases containing the lexical information found in print dictionaries, as well as potentially additional, richer linguistic information about words and their relations to one another. Publishers see the potential to generate several variant printed dictionaries from such a database, and researchers foresee their use in linguistics research and automated language processing tasks (natural language understanding, machine translation, etc.).

In order to create a sophisticated dictionary database, it is necessary to find a model capable of representing the lexical information typically found in printed dictionary entries. However, the textual properties of dictionary entries and their complex structure make it difficult to determine which kind of model will be adequate (see Boguraev *et al.,* 1990). In this paper, we show that current models for representing text cannot capture the complex structure of dictionaries in a form that facilitates sophisticated retrieval. On the other hand, we show that existing database models are inadequate to represent the rich information in dictionary entries. We therefore propose that a new model must be developed to represent dictionaries, and describe the basic features such a model will require. We then demonstrate the applicability of our model to the data in printed monolingual dictionaries in English and French.

The model we have developed is intended to be flexible enough to represent printed dictionaries with different structures in a common database format. This has demanded consideration of the nature, structure, and interrelations of lexical information, independent of its manifestation in any particular dictionary, as well as consideration of the uses to which the database will be put. We see the development of a common format for the lexical information found in print dictionaries as the basis or core of a more generalized lexical entry format, capable of representing not only the lexical information explicit in printed dictionaries, but also other information which may be derived from print dictionary entries or added to it.

The work described in this paper has been carried out in the context of a joint project of the Department of Computer Science at Vassar College and the Groupe Représentation et Traitement des Connaissances of the Centre National de la Recherche Scientifique (CNRS), which is concerned with the construction and exploitation of a large lexical data base of English and French. At present, the Vassar/CNRS data base includes, through the courtesy of several editors and research institutions, several English and French dictionaries (the *Collins English Dictionary,* the *Oxford Advanced Learner's Dictionary,* the *COBUILD Dictionary,* the *Longman's Dictionary of Contemporary English*, the *Webster's 9th Dictionary,* and the *ZYZOMYS* CD-ROM dictionary from Hachette Publishers) as well as several other lexical and textual materials (*Roget's Thesaurus,* the *Brown Corpus of American English,* the *London-Oslo-Bergen Corpus,* a corpus of French texts and novels, the CNRS *BDLex* data base, the *MRC Psycholinguistic Data Base,* etc.). A broad description of the database and the activities within the Vassar/CNRS project appears in Véronis *et al.* (1990).

2. Tagged text models

Dictionaries were first realized electronically as typesetter's tapes for the purposes of publishing. These tapes subsequently became available to the research community and have been processed extensively in order to extract lexical information (see, for instance, Amsler, 1980; Calzolari, 1984; Chodorow *et al.,* 1985; Markowitz *et al.,* 1986; Byrd *et al.,* 1987; Slator and Wilks, 1987; Véronis and Ide, 1990; Ide and Véronis, 1991). In this form, a dictionary exists as a strictly linear text stream interspersed with *markup*, that is, *tags* that signal the beginning and end of any of several fields of information. We use the term *tagged text model* to refer to this representation of the dictionary as a text stream comprising content and interspersed markup.

In typesetter's tapes, markup is used to signal a font shift or the presence of special characters, etc., corresponding to the rendering of the dictionary in printed form (figure 1). Markup of this kind is called *procedural,* because it specifies the procedure (e.g., shift to italic) to be performed when a tag is encountered in linear processing, rather than providing an indication of content of the field (see Coombs, *et al.,* 1987). Although typographic codes are to some extent indicative of field content (for example, part of speech may always be in italics in a given dictionary), a straightforward, one-to-one mapping between typographic codes and content clearly does not exist (it is common for several other items, such as semantic field, usage, register, geographical information etc., to be rendered in italics as well). Positional information can be coupled with typographic tags to determine content, but a complex analysis of entry format, which may or may not yield a definitive mapping due to ambiguities, is required. Information retrieval from a dictionary in this form is obviously costly, if possible at all.

```
*Cgin*E*S1*E (d*3T*3Fn) *Fn. *%brew *5Q*A1. *Ean alcoholic drink obtained
by distillation and rectification of the grain of malted barley, rye, or
maize, flavoured with juniper berries. *%brew *5Q*A2. *Eany of various
grain spirits flavoured with other fruit or aromatic essences: *Fsloe
gin. *%brew *5Q*A3. *Ean alcoholic drink made from any rectified spirit.
*5Q*5HC18: shortened from Dutch *Fgenever *Ejuniper, via Old French form
Latin *Fj=u-niperus*Gjuniper*E*5I<
```

Figure 1. Typographical markup (*Collins English Dictionary*)

In a *descriptive* markup scheme, tags provide an indication of the content of the fields they delimit rather than the printed rendering. For instance, instead of tags for italics, bold, etc., tags indicate *headword, part of speech, pronunciation,* etc. (figure 2). There have been a number of efforts to devise descriptive markup schemes for monolingual dictionaries (see, for instance, Tompa, 1989) and to translate the procedural markup of typesetter's tapes into descriptive markup (see Harié, 1989, and Bogureav and Neff, 1991). More recently, a preliminary common set of descriptive tags for encoding mono- and bi-lingual dictionaries has been proposed (Amsler and Tompa, 1988) which was subsequently incorporated into the international Text Encoding Initiative's guidelines for encoding machine readable textual and linguistic data (Sperberg-McQueen and Burnard, 1990).

```
<ent h=gin hn=2><hdw>gin</hdw><pr><ph>dZIn</ph></pr>
<hps ps=n cu=U><hsn><def>colourless alcoholic drink distilled from grain
or malt and flavoured with juniper berries, often drunk with tonic water,
and used in many kinds of cocktail</def></hsn></hps></ent>
```

Figure 2. Descriptive markup (*Oxford Advanced Learner's Dictionary*)

The use of descriptive markup enables the retrieval of information by content category from dictionaries in a linear text format. Software to perform such operations on descriptively tagged text exists (for example, PAT; Gonnet and Tompa, 1988). Retrieval software of this kind regards markup as strings of characters embedded in text and basically performs sophisticated string searches. However, although such software provides powerful searching capabilities, it is nonethless limited for contextual searching. Searches which involve elements whose relationship is embodied in the structure of the dictionary entry can become prohibitively complex. For example, to find the part of speech for sense 4 of a given word, one needs to access information typically appearing prior to the listing of sense definition texts, and associated with a group of definitions by virtue of its location. To retrieve this information (given that one has identified sense 4 as the object of interest), some analysis of the surrounding text of the entry is required. This can be costly, and potentially very complex or impossible for complicated retrieval requests.

3. Relational database models

Database models for dictionaries have been proposed, primarily for the purposes of research in computational linguistics. Database models have been less popular with publishers and lexicographers, who have traditionally mistrusted such models as too simplistic and/or rigid to allow the editorial freedom lexicographers desire when creating dictionaries (Tompa, 1989). The most common database models are *relational models.* A relational database consists of a set of *relations* between *entities*. Each role in that relation is called an *attribute*. Conceptually, a relation is a *table* whose columns correspond to *attributes*, and each row specifies all the values of attributes for a given entity.

The most common relational model is the *normalized* relational model, in which attributes have only *atomic values*--that is, values which, from the database system's point of view, cannot be decomposed. In other words, each row-to-column intersection contains one, and only one, value. Recently, *unnormalized* relational models have been proposed, in which values are not necessarily atomic but may have some internal structure. Both normalized and unnormalized models have been proposed for representing dictionaries.

3.1. Normalized relational models

Normalized relational models have been suggested for representing dictionary information (Nakamura and Nagao, 1988; Fontenelle and Vanandroye, 1989). In these schemes, the dictionary is represented by a set of relations, each of which includes attributes such as grammar codes, definitions, examples, etc. Figure 3 gives the definition of "abandon" from the *Longman's Dictionary of Current English (LDOCE);* figure 4, expanded from Nakamura and Nagao, shows the tabular representation of the same entry. Note that certain information, such as the *LDOCE* semantic "box codes", appears only in the machine readable version of the dictionary, and it therefore appears in the database even though absent from the printed version.

a·ban·don[1] /@'b&nd@n/ *v* [T1] **1** to leave completely and for ever; desert: *The sailors abandoned the burning ship*. **2** to leave (a relation or friend) in a thoughtless or cruel way: *He abandoned his wife and went away with all their money*. **3** to give up, esp. without finishing: *The search was abandoned when night came, even though the child had not been found*. **4** (*to*) to give (oneself) up completely to a feeling, desire, etc.: *He abandoned himself to grief | abandoned behaviour.* -- **~ment** *n* [U].

abandon[2] *n* [U] the state when one's feelings and actions are uncontrolled; freedom from control: *The people were so excited that they jumped and shouted with abandon / in gay abandon.*

Figure 3. Definition of 'abandon' from *LDOCE*

HW	PS	DN	DF
abandon	v	1	to leave completely and for ever
abandon	v	1	desert
abandon	v	2	to leave (a relation or friend) in a thoughtless or cruel way
abandon	v	3	to give up, esp. without finishing
abandon	v	4	to give (oneself) up completely to a feeling, desire, etc.
abandon	n	0	the state when one's feelings and actions are uncontrolled
abandon	n	0	freedom from control

HW	PS	DN	SP
abandon	v	1	The sailors abandoned the burning ship
abandon	v	2	He abandoned his wife and went away with all their money
abandon	v	3	The search was abandoned when night came, even though the child had not been found
abandon	v	4	He abandoned himself to grief
abandon	v	4	abandoned behaviour
abandon	n	0	The people were so excited that they jumped and shouted with abandon/in gay abandon

HW	PS	DN	GC	BC
abandon	v	1	T1	----H----T
abandon	v	2	T1	--D-H----H
abandon	v	3	T1	----H----T
abandon	v	4	T1	----H----H
abandon	n	0	U	----T-----

HW = headword
PS = part of speech
DN = definition number
DF = definition text
SP = example
GC = grammar code
BC = LDOCE "box" code

Figure 4. Tables for 'abandon' in *LDOCE* database

This example is derived from a small, simple learner's dictionary with a straightforward internal structure (no deep nesting of senses, etc.), and several pieces of information from the entry text (for example, pronunciation, run-ons, cross-references) have been omitted from the database. However, even this simplified case shows that the relational model poses several problems for representing dictionary entries. The most obvious problem is that the information contained in an entry must be split across several tables, thus fragmenting the view of the data. The more complex the data, the more tables are required. To enable the user to view information from more than one table, it is necessary to create a new table from the Cartesian product of these tables, and the resulting table contains an enormous amount of redundant information. In the *abandon* example above, it is already clear that since the relations are not elementary (that is, headword, part-of-speech, and sense definition all appear in the same table), there is considerable duplication (in particular, headword and part of speech are repeated several times). This difficulty arises from the fact that in the normalized relational model, the values of attributes cannot be complex objects. It is therefore difficult to represent highly structured information such as that contained in dictionary entries. Other problems arise from the fact that the relational model defines a fixed number of attributes for each entity in the database. This is extremely wasteful for representing dictionaries, which vary greatly in the kinds and amount of information included. For example, one entry may include only pronunciation, part of speech, and definition, while another includes examples, synonyms, cross-references, domain information, geographical information etc.

3.2 Unnormalized relational models

Neff, Byrd, and Rizk (1988) describe an organization for a lexical database (the IBM LDB) that addresses many of the problems inherent in the relational model (see also Calzolari *et al.,* 1990). They propose a shallow hierarchy of attribute-value pairs to represent a dictionary entry, in which the kinds and number of attributes associated with a given sense can vary (Figure 5). Although the authors do not make this claim, their model seems to bear some similarity to unnormalized relational models, in which, unlike the normalized relational model, attribute values may have internal structure. Unnormalized relations with such attributes are also called "nested relations" or "NF squared (Non First Normal Form) relations". There is a growing body of research into the properties of these relations, since normalized relations can be cumbersome for representing non-traditional data such as CAD/CAM, statistics, text, graphics, etc. An algebra and calculus have been proposed for nested relations (Roth *et al.,* 1988), and a few database systems have been developed using the unnormalized model (e.g., AIM-P at IBM Heidelberg; see Pistor and Traunmueller, 1986).

In the IBM LDB model, information in an entry can be represented by a tree (see Figure 5). This adds the capability to better express the fundamentally hierarchical structure of information in dictionaries.

However, in both the normalized and IBM LDB models, the internal nesting of senses is misrepresented. For instance, the two senses of the verb labeled "1" in Figure 4 are in fact two sub-senses of the first sense given in the entry; this organization reflects the fact that they are more closely related to each other than to senses 2, 3, and 4, but the tabular format obscures this fact. The problem is the same with the IBM LDB, which does not provide for arbitrarily deep sense nesting. These approaches can lose significant information about relations among senses: some dictionaries take the grouping and nesting of senses several levels deep in order to distinguish finer and finer grains of meaning. The Hachette *Zyzomys* CD-ROM dictionary (which is close to the paper dictionary *Le Dictionnaire de Notre Temps*), for instance, distinguishes up to five levels in an entry. Figure 6 shows that in this dictionary "valeur" has two fundamental senses: (A) *value* as *merit* ; and (B) *value* as price. Going deeper, we see that sense A subdivides into two main subcategories: (I) merit of an individual; and (II) the subjective worth of an object. Sense A.I subdivides further into two more subcategories: (1) merit of a person based on general qualities; and (2) bravery or valor, which in turn forms a part of the compound "croix de la valeur militaire", a French military decoration.

```
entry
+-hdw: abandon
|
+-superhom
| +-word: abandon
| +-print_form: a.ban.don
| +-hom_number: 01
| |
| +-pronunciatio
| | +-primary
| |   +-pron_string: E"b∑ndEn
| |
| +-syncat: v
| +-g_code_field: T1
| |
| +-sense_def
| | +-sense_no: 1
| | +-subj_code: ....
| | +-box_code: ....H....T
| | |
| | +-defn
| | | +-def_string: to leave completely and for ever; desert
| | |
| | +-example
| |   +-ex_string: The sailors abandoned the burning ship
...
...
| +-run_on
|   +-sense_link: 01
|   +-derivative: abandonment
|   +-d_syncat: n
|   |
|   +-d_code
|     +-g_code_field: U
|
+-superhom
  +-word: abandon
  +-print_form: abandon
  +-hom_number: 02
  +-syncat: n
  +-g_code_field: U
  |
  +-sense_def
    +-sense_no: 0
    +-subj_code: ....
    +-box_code: ....T.....
    |
    +-defn
    | +-def_string: the state when one's feelings and actions...
    |
    +-example
      +-ex_string: The people were so excited that they jumped...
```

Figure 5. IBM LDB format for 'abandon' in the *LDOCE*

Flattening the structure described above into a tabular form or the IBM LDB format would create substantial redundancy and obscure the derivational relations captured in the nested arrangement. It would also destroy the upper level senses as identifiable entities, since, for example, there would no longer be a way to address *valeur A* or *valeur B* as a whole. It is important to retain entities corresponding to each sense level in the database, in order, for example, to enable a cross-reference to a sense group (e.g., *valeur A*). This would also enable generating a beginner's dictionary including only the top-level senses, as well as an expert dictionary containing the finest sub-divisions of meaning, from the same database. In addition, if the database is used for automated language processing, varying degrees of precision in distinguishing senses may be required: broad sense distinctions may be enough to deal with prepositional phrase attachment, whereas machine translation may demand finer sense distinctions.

valeur [valœR] n. f. **A. I. 1.** Ce par quoi une personne est digne d'estime, ensemble des qualités qui la recommandent. (V. mérite). *Avoir conscience de sa valeur. C'est un homme de grande valeur.* **2.** Vx. Vaillance, bravoure (spécial., au combat). *"La valeur n'attend pas le nombre des années"* (Corneille). ◊ *Valeur militaire (croix de la):* décoration française...
…
II. 1. Ce en quoi une chose est digne d'intérêt. *Les souvenirs attachés à cet objet font pour moi sa valeur.* **2.** Caractère de ce qui est reconnu digne d'intérêt...
…
B. I. 1. Caractère mesurable d'un objet, en tant qu'il est susceptible d'être échangé, désiré, vendu, etc. (V. prix). *Faire estimer la valeur d'un objet d'art...*

Figure 6. Part of the definition of 'valeur' in Hachette *Zyzomys*

In order to properly handle the multi-level nesting of senses, it would be necessary to define a recursive attribute (i.e., the value of a sense can be a set of senses), However, this is not allowed in the IBM LDB model, nor, in general, in unnormalized relational models.

Another problem with the IBM LDB model is that certain kinds of information are specified to appear at only certain points in the tree. For example, pronunciation is a feature associated with a particular homograph, and not with individual senses of that homograph. The fixed format does not provide a built-in exception mechanism, to enable an "override" of an attribute value for a particular sense or sub-sense. Sense 3 of the word "conjure" in the *Oxford Advanced Learner's Dictionary (OALD),* for example, has a different pronunciation from the other senses in the entry (figure 7). The only way to solve this in the IBM LDB format is to enable a pronunciation attribute at both the homograph and sense levels, which creates a problem of consistency since the same information is associated with different objects. Most attributes would likely have to be treated this way (since in exceptional cases, senses have variants, special spellings, etc.). If anything can appear as an attribute of any object, the model becomes ill-defined and the notion of hierarchy is undermined.

conjure /'kVndZ@(r)/ *vt,vi* **1** [VP2A,15A] do clever tricks which appear magical... **2** [VP15B] **~ *up,*** cause to appear as if from nothing... **3** /k@n'dZU@(r)/ [VP17] (formal) appeal solemnly to...

Figure 7. Definition of 'conjure' in *OALD*

Finally, the lack of flexibility of the IBM LDB and similar formats demands a different template for each dictionary. This makes the merging of information, the use of common software, etc., more difficult and costly. We seek a format which can generalize across dictionaries in order to avoid these problems.

4. Outline of a model for representing dictionaries

The discussion in section 3 makes it clear that previously proposed models for representing dictionaries need to be substantially modified. In this section we propose a model that answers many of the general objections posed above. For lack of space, we present here only the major structural features of the model; we do not provide fine details such as a comprehensive listing of all necessary attributes. However, we believe that the difficult problem in developing a model for representing dictionaries consists more in devising an appropriate structural framework than in fully specifying all necessary attributes. But before developing a structural framework, it is necessary to consider in detail the nature of the information that we model, and in particular, to determine what the fundamental entities or objects in the database are.

4.1. Basic objects

Printed dictionaries are organized according to individual *entries,* each of which consists of a key or headword coupled with a number of senses. Entries are arranged in alphabetical order according to the headword key. However, the choice of entries is, as lexicographers themselves admit, somewhat arbitrary, which the differences among dictionaries (and even different editions of the same dictionary) make especially clear.

First of all, lexicographers make *presentation choices* based on criteria such as intended audience. For example, in one dictionary, all senses of a given orthographic form with the same etymology will be grouped in a single entry, regardless of part of speech; whereas in another, different entries for the same orthographic form are given if the part of speech is different. The *Collins English Dictionary (CED)*, for instance, has only one entry for *abandon,* including both the noun and verb forms, but the *LDOCE* gives two entries for *abandon,* one for each part of speech (see above). Similarly, in some dictionaries, related entries (e.g., phrasal verbs, compounds) are given within the entry of the main word to which they are related, while in others they appear as separate entries: for example, the *OALD* gives related entries such as *bear down* and *bear skin* within the entry for *bear,* whereas the *CED* has separate entries for each.

Second, the *physical arrangement* of dictionaries often dictates the choice of entries. For instance, when variant or inflected forms of a word exist, the lexicographer will typically choose one to serve as the key to the entry in which the definition text(s) appears, and, for reasons of space, list the others within the same entry rather than giving each an entry of its own. However, in cases where the variant or inflected form is alphabetically distant from the key, it is given in a separate entry as well (usually, including only a cross-referential text such as "variant form of ..."). Related entries (derivatives, compounds, etc.) which are alphabetically distant are treated the same way. This is obviously done because the dictionary user would look in a *physically* separate part of the dictionary for the variant or inflected form, which, if not given there, would be unretrievable. For example, in the entry for *fakir* in the *CED, faqir* and *fakeer* are given as variant spellings; however, only *faqir* also has a separate (cross-referential) entry, because its alphabetic position in the dictionary is several pages after that of *fakir.* Thus, in order to facilitate retrieval in printed dictionaries, a compromise must be struck between logical groupings and alphabetic distance, the latter of which may be measured in terms of linguistically factors as irrelevant as page layout.

The impact of space considerations on the organization of entries is pervasive in dictionaries. For example, most compounds should logically be accessible by looking under any of their components--the compound *German measles,* for instance, should be retrievable by looking under either *German* or *measles.* To save space, lexicographers have three options: store the definition of *German measles* under *German* and include a cross reference to this definition under *measles;* store the definition under *measles* and include a cross-reference under *German;* or make *German measles* a separate entry with cross-references under both *German* and *measles.* It is interesting to note the inconsistencies that such practice leads to in printed dictionaries: the *OALD* gives the definition of *German measles* under *German,* but gives no cross-reference under *measles*; the *OED* gives the full definition under both *German* and *measles*, but gives a different definition in each case; and the *LDOCE* and *CED* both have a separate entry for *German measles*, with a cross-reference under *measles* but not under *German* (which is probably felt to be close enough in alphabetic order).

In a database, presentation constraints, physical layout, and space considerations are no longer relevant. We can imagine that from the same database, a number of physical variants of the dictionary could be produced. For example, senses can be displayed in any desired configuration: in isolation, in groups on the basis of common etymology, according to part of speech, etc. Similarly, definitions of variant and inflected forms and related entries can be displayed under their own entries as well as under any related or component word: for example, the ideal situation in the *German measles* example above might be to *store* the definition in one place, but create links which enable the definition to be *displayed* under *German, measles,* and *German measles.* These are only a few examples, intended to make clear the difference between the presentation or display of dictionary data, and the information used to generate that display. Previous models have not made this distinction, and as a result they have been built around the concept of an *entry* used in the production of printed dictionaries. Because of this, they have (unnecessarily) duplicated the organizational problems in printed dictionaries. For example, they duplicate the varying treatment within the same dictionary of elements such as related words, which, depending on physical considerations, sometimes appear within another entry and sometimes appear as separate entries. Therefore, the same objects are represented differently in the database, and no uniform procedure can be used to access them.

We believe it is necessary to ignore the traditional concept of en entry in the design of a dictionary database. In our model, the sense is the basic object. This choice was made because, logically, all attributes (orthographic form, pronunciation, part of speech, etymology, etc.) pertain to senses. The physical organization of the dictionary often obscures this fact, especially in cases where an attribute appears at the beginning (or end) of an entry because it applies to all senses. Nonetheless, the fact that an attribute such as pronunciation can be attached to a single sense (as in the "conjure" example given above) demonstrates that this attribute applies to senses and not to entries. On reflection, it is clear that this is true for all attributes that are typically found in a dictionary.

Figure 8 shows the main attributes in our model and their relationships. As in the IBM LDB model, attributes can be either *terminal* (in lower case) or *non-terminal* (in upper case). Terminal attributes (`sn`, `def`, `time`, etc.) contain specific values (number, string of characters, etc.). Non-terminal attributes (`SENSE`, `FORM`, `GRAM`, etc.) contain only other attributes (for example, `FORM` contains `orth`, `pron`, etc.). However our model differs from the IBM LDB model in one important way: our scheme allows the recursive embedding of non-terminal attributes. This changes dramatically the properties of the model, as explained in the next section. Attributes may have *null values* (when information does not apply or exist--for example, when no example or etymology is given) and *default values* (for example, when no geographic domain is specified, the default would be British in British dictionaries). Most attributes can appear one or more times at any allowable point.

```
SENSE:
  FORM              /* form group */
  GRAM              /* grammar group */
  sn                /* sense number */
  def               /* definition text */
  time              /* archaic, etc. */
  geo               /* geographic area (for sense) */
  SEMANT            /* semantic information group */
  EXAMPLE           /* example group */
  XREF              /* cross-reference group */
  RELATED           /* related object group */
  ETYM              /* etymological group */
  SENSE

FORM:
  orth              /* orthography */
  pron              /* pronunciation */
  hyph              /* hyphenation */
  geo               /* geographic area (for form) */
  FORM

GRAM:
  pos               /* part of speech */
  subc              /* subcategorization */
  gend              /* gender */
  numb              /* number */

SEMANT:
  reg               /* formal, informal, etc. */
  dom               /* chemistry, nautical, jewelry, etc. */

EXAMPLE:
  text              /* text of example */
  auth              /* author of example */
  date              /* date of example */

XREF:
  type              /* see also; antonym; synonym */
  orth              /* orthographic form of object referred to */
  sn                /* sense number of object referred to */

RELATED:
  type              /* compound, derivative, etc. */
  orth              /* orthographic form of object referred to */

ETYM:
  ...               /* etymological attributes not yet worked out */
```

Figure 8. Grammar of most common attributes

Occasionally, the addition of some dictionary-specific attributes is required (e.g., certain box codes for the *LDOCE*), but in general the attributes in our model are the same for all monolingual dictionaries. More importantly, the grammar for the common attributes remains the same across dictionaries, which constitutes a significant difference between ours and previously proposed models.

4.2 Relations between objects

Within our model there are three kinds of relations that can exist between sense objects:

1) *No relation.*

2) *Linkage.* In our model, related words, phrases, and compounds are separate objects. A link is created between the source(s) and the derivative. Linkage is also used for cross-references. figure 9 shows how related entries and cross-references are linked, using an example based on defintions in the *LDOCE*. This scheme assures that all related objects are represented in the same way within the database. Note that the link from *German* to *German measles* does not appear in the *LDOCE*, but adding such a link when creating the database is trivial.

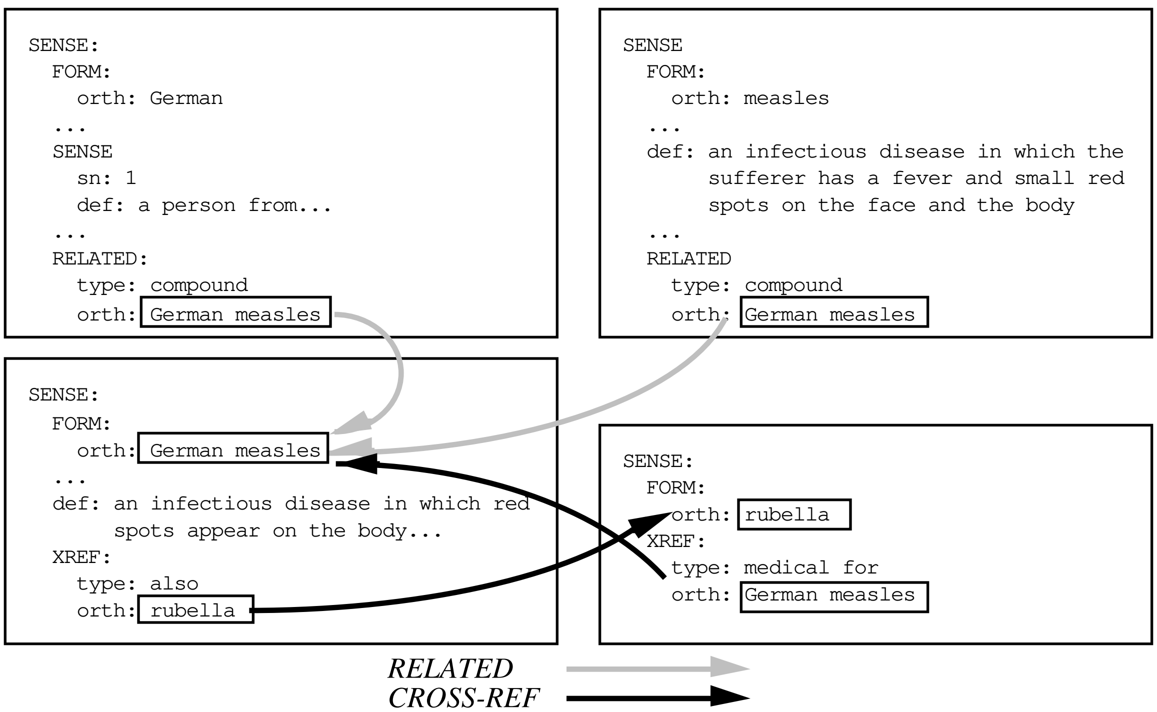

Figure 9. Related objects and cross-references

3) *Self-embedding.* This provides for the nesting of senses within senses, thus retaining the clear identification of different sense levels. Additionally, an *inheritance mechanism* is provided which enables the factoring of information, thus eliminating redundancy. For example, if the same part of speech applies to two sub-senses, it can be factored out as in Figure 10. Thus, although no value for part of speech is specified for sub-senses A, A.I, A.I.1., the inheritance mechanism will automatically ensure that the query "what is the part-of-speech of sense A.I.1?" will return the value *n.* Inheritance applies over any depth of nesting.

```
SENSE:
  FORM:
    orth: valeur
    pron: valœR
  GRAM:
    pos: n
    gend: f
  SENSE:
    sn: A
    SENSE:
      sn: I
      SENSE:
        sn:  1
        def: Ce par quoi une personne est digne d'estime...
        XREF:
          orth: mérite
        EXAMPLE:
          text: Avoir conscience de sa valeur.
        EXAMPLE:
          text: C'est un homme de grande valeur.
      SENSE:
        sn:  2
        time: Vx
        def: Vaillance, bravoure (spécial., au combat).
        EXAMPLE:
          text: La valeur n'attend pas le nombre des années
          auth: Corneille
        RELATED:
          orth: croix de la valeur militaire
      ...
    SENSE:
      sn: II
      SENSE:
        sn:  1
        def: Ce en quoi une chose est digne d'intérêt.
        EXAMPLE:
          text: Les souvenirs attachés à cet objet...
      SENSE:
        sn:  2
        def: Caractère de ce qui est reconnu digne...
  ...
  SENSE:
    sn: B
    SENSE:
      sn: I
      SENSE:
        sn:  1
        def: Caractère mesurable d'un objet...
        XREF:
          orth: prix
        EXAMPLE:
          text: Faire estimer la valeur d'un objet d'art.
      ...
```

Figure 10. Self-embedding of senses ('valeur' in *Zyzomys*)

A different value can be specified for an attribute at an inner level, in which case two possible mechanisms can apply:

1) the value specified at the outer level is *overridden* by the new value, for this sub-sense and any sub-senses nested in it. For example, in figure 11, the pronunciation specified at the outermost level is inherited by all senses except sense 3, where it is overriden. This provides a convenient exception mechanism.

2) the actual value is some *combination* of all the values found along the path of nested senses--e.g., in Figure 10, the actual sense number for the sense "vaillance" is the concatenation of all sense numbers in the path of senses to it, that is, *A.I.2*.

conjure /'kVndZ@(r)/ *vt,vi* **1** [VP2A,15A] do clever tricks which appear magical... **2** [VP15B] **~ *up,*** cause to appear as if from nothing... **3** /k@n'dZU@(r)/ [VP17] (formal) appeal solemnly to... [OALD]

```
SENSE:
  FORM:
    orth: conjure
    pron: "kVndZ@(r)                          /* pron. for all senses */
  GRAM:
    pos: v
    subc: tr
    subc: intr
  SENSE:
    sn: 1
    GRAM:
      gcode: VP2A
      gcode: VP15A
    def: do clever tricks which appear magical...
  SENSE:
    sn: 2
    GRAM:
      gcode: VP15B
    RELATED:
      orth: conjure up
  SENSE:
    sn: 3
    FORM:
      pron: k@n"dZU@(r)   /* overrides "kVndZ@(r) for sense 3 only */
    GRAM:
      gcode: VP17
    SEMANT:
      reg: formal
    def: appeal solemnly to...
...
```

Figure 11. Overriding of values

Attributes other than SENSE may be self-embedded--in particular, FORM. This allows for the grouping of variant forms, as demonstrated in figure 12.

distil *or U.S.* **distill** (dIs'tIl) ... [CED]

```
SENSE:
  FORM:
    orth: distil
    pron: dIs'tIl            /* pronunciation is shared */
    FORM:
      geo:  U.S.             /* overrides default (Brit) */
      orth: distill          /* overrides 'distil' */
...
```

alumnus (@'l^mn@s) *or* (*fem.*) **alumna** (@'l^mn@) *n., pl.* **-ni** (-naI) *or* **-nae** (-ni:) ... [CED]

```
SENSE:
  FORM:
    orth: alumnus
    pron: @"l^mn@s
    FORM:
      numb: pl                 /* overrides default number (sing) */
      orth: alumni             /* overrides 'alumnus' */
      pron: @"l^mnaI           /* overrides '@"l^mn@s' */
    FORM:
      gend: fem                /* overrides default gender (masc) */
      orth: alumna             /* overrides 'alumnus' */
      pron: @"l^mn@            /* overrides '@"l^mn@s' */
      FORM:
        numb: pl               /* overrides default number (sing) */
        orth: alumnae          /* overrides 'alumna' */
        pron: @"l^mni:         /* overrides '@"l^mn@' */
...
```

Figure 12. Self-embedding of the FORM attribute

Another example of our model is given in figure 13 for the entry *abandon* in *LDOCE,* which was discussed in section 3. Note that the representation of *abandon* from the *LDOCE* groups the two homographs together as one object. This enables the factoring of information over both homographs, which eliminates, for instance, a re-specification of the orthographic form. More importantly, it allows hyphenation and pronunciation to be factored over both homographs: interestingly, the *LDOCE* gives a separate entry for each part of speech, but gives information about hyphenation and pronunciation only in the entry for the first homograph. This shows again that entries are an artifact of printed presentation and do not entirely reflect logical structure. The IBM LDB representation of the *LDOCE* entry for *abandon* loses the information about hyphenation and pronunciation for the second homograph, since there is no provision for the factoring of information in this scheme. The only solution in that model would be to repeat the information for the second homograph.

```
SENSE:
  FORM:
    orth: abandon
    hyph: a.ban.don                     /* apply to both homographs */
    pron: @"b&nd@n
  SENSE:
    GRAM:
      pos: v
      gramc: T1
    SENSE:
      sn: 1
      SEMANT:                     /* subject code is ---- by default */
        boxc: ----H----T
      def: to leave completely and for ever
      def: desert
      EXAMPLE:
        text: The sailors abandoned the burning ship
    ...
    RELATED
      link: abandonment                         /* has its own entry */
  SENSE:
    GRAM:
      pos: n
      gramc: U
    SENSE:
      SEMANT:
        boxc: ----T-----
      def: the state when one's feelings and actions...
      EXAMPLE:
        text: The people were so excited that they jumped...
```

Figure 13. Representation of 'abandon' in *LDOCE*

We have implemented a prototype of a database system following the scheme outlined here together with a number of retrieval functions, using the database language GRIFFON (Le Maître, 1988). This implementation has demonstrated the validity of the model. A full-scale implementation is currently being developed, using the object-oriented O2 database management system (Lecluse and Richard, 1989). Object-oriented databases seem well-suited to represent dictionaries, since they allow for highly structured objects by providing complex built-in type constructors such as lists and sets, as well as the construction of new types, and in particular, recursive types. The underlying principle of the object-oriented approach is to eliminate computer-based concepts such as records and fields (the fundamental concepts in the relational model), and enable the user to deal with higher-level concepts that correspond more directly to the real world objects the database represents. Objects within the database, together with all of the attributes (and even procedures for manipulating these attributes) associated with them, are considered as wholes, whereas in relational models, objects do not exist as wholes but are instead split across the various relations defined in the database.

4.3 From printed dictionary to database and back

Our goal here is not to find a means to represent existing printed dictionaries. Our model is intended to represent *electronic dictionary databases,* which can be displayed on a screen, printed in (possibly) several variant forms, or used in research as a source of lexical information for linguistic studies and automated language processing. We currently translate dictionaries in the form of typesetter's tapes into databases because we are not able to develop such databases from scratch. In the future, this translation will be unnecessary, since we foresee that dictionaries will ultimately exist and evolve in database form. Printed versions will likely be generated from such databases.

Translation from typesetter's tapes to database form is problematic, as pointed out by Boguraev and Neff (1991), due primarily to ambiguities and inconsistancies within the printed dictionaries themselves. For instance, in the *CED*, the entry *Canopic jar, urn* or *vase* must be interpreted as (*Canopic jar*) or (*Canopic urn*) or (*Canopic vase*), whereas the entry *Junggar Pendi, Dzungaria,* or *Zungaria,* which has the same structure, must be interpreted as (*Junggar Pendi*) or (*Dzungaria*) or (*Zungaria*). Because of the inconsistency, fully automated procedures cannot determine the appropriate interpretations. At the same time, in other cases it is difficult to regenerate the exact form of the original printed version from the database. For example, the *ZYZOMYS* dictionary gives the headword *curieux, -euse,* which is expanded when stored in the database to the forms *curieux* and *curieuse.* However, the *ZYZOMYS* also gives (inconsistently) *mystérieux, -ieuse.* The only way to know which form appears in the original would be to preserve the original text in the database, as Calzolari *et al.* (1990) propose. We see no motivation for retaining information about printed formats in the database, since it is not clear that an exact reproduction of the original is useful. Further, algorithms for regenerating the original can be devised in cases where layout conventions are adhered to consistently; it seems unproductive to retain information about inconsistencies in dictionary format, which publishers themselves would likely be pleased to eliminate. We believe that printed versions of dictionaries produced algorithmically from databases will likely fulfill most publishers' needs in the future.

5. Conclusion

We have outlined a scheme for representing electronic dictionaries which departs from previously proposed models in significant ways. In particular, it allows for a full representation of sense nesting and defines an inheritance mechanism which enables the elimination of redundant information. The model provides flexibility which seems able to handle the varying structures of different monolingual dictionaries. Our model may therefore be applicable to a diversity of uses of electronic dictionaries, ranging from research to publication. Lexicographers, in particular, have preferred tagged text to lexical databases because they fear a loss of editorial freedom. However, we believe that a flexible model such as the one outlined here can both serve the needs of lexicographers with minimal constraints, and at the same time provide powerful retrieval capabilities and the potential to display the information in a variety of forms and formats.

A number of open problems remain for fully specifying the structure and elements of dictionaries. For example, we have not addressed the problems of phrasal elements (such as discontinuous verb phrases and cross-reference phrases embedded in definition or example text), etymologies (which are themselves complex structured text), etc. Further, it is necessary to test our model across a wide range of monolingual dictionaries in order to ascertain, first, its generality and, second, the exact scope and nature of remaining difficulties.

References

AMSLER, R. A. (1980). The structure of the Merriam-Webster Pocket Dictionary. *Ph. D. Dissertation,* University of Texas at Austin.

AMSLER, R. A., TOMPA, F. W. (1988). An SGML-based standard for English monolingual dictionaries. *Proceedings of the 4th Annual Conference of the UW Centre for the New Oxford English Dictionary,* Waterloo, Ontario, 61-80.

BOGURAEV, B., BRISCOE, E., CARROLL, J., COPESTAKE, A. (1990). Database models for computational lexicography, presented at *EURALEX,* Malaga, Spain.

BOGURAEV, B., NEFF, M. S. (1991). From machine readable dictionaries to lexical databases. Forthcoming in *International Journal of Lexicography.*

BYRD, R. J., CALZOLARI, N., CHODOROW, M. S., KLAVANS, J. L., NEFF, M. S., RIZK, O. (1987) Tools and methods for computational linguistics. *Computational Linguistics,* 13, 3/4, 219-240.

CALZOLARI, N. (1984). Detecting patterns in a lexical data base. *Proceedings of the 10th International Conference on Computational Linguistics,* COLING'84, 170-173.

CALZOLARI, N., PETERS, C., ROVENTINI, A. (1990). *Computational Model of the Dictionary Entry: Preliminary Report,* ACQUILEX, Esprit Basic Research Action No. 3030, Pisa, Italy.

CHODOROW, M. S., BYRD. R. J., HEIDORN, G. E. (1985). Extracting semantic hierarchies from a large on-line dictionary. *Proceedings of the 23rd Annual Conference of the Association for Computational Linguistics,* Chicago, 299-304.

COOMBS, J. H., RENEAR, A. H., DEROSE, S. J. (1987). Markup systems and the future of scholarly text processing. *CACM,* 30:11, 933-47.

FONTENELLE, T., VANANDROYE, J. (1989). Retrieving ergative verbs from a lexical database, ms., English Department, University of Liege.

GONNET, G., TOMPA, F. W. (1988). Mind your grammar: a new approach to modelling text. *Proceedings of the 13th. Conference on Very Large Data Bases,* VLDB'87, Brighton, England, 339-346.

HARIÉ, S. (1989). *Analyse automatique d'un dictionnaire en vue de la constitution d'une base de données lexicale.* Mémoire de DEA, Université Aix-Marseille III.

IDE, N.M., VÉRONIS, J. (1990). Refining taxonomies extracted from machine-readable dictionaries, *ALLC/ACH'90 Conference,* Siegen, Germany [to appear in selected proceedings, Oxford University Press]

LECLUSE, C., RICHARD, P., (1989). The O2 database programming language, *Proceedings of the 15th VLDB Conference,* Amsterdam, August, 1989.

LE MAÎTRE, J. (1988). Le langage de manipulation de bases de données GRIFFON, *JISI'88*, Tunis, 15-17 Avril 1988.

MARKOWITZ, J., AHLSWEDE, T., EVENS, M. (1986). Semantically significant patterns in dictionary definitions. *Proceedings of the 24rd Annual Conference of the Association for Computational Linguistics,* New York, 112-119.

NAKAMURA, J., NAGAO, M. (1988). Extraction of semantic information from an ordinary English dictionary and its evaluation. *Proceedings of the 12th International Conference on Computational Linguistics,* COLING'88, 459-464.

NEFF, M. S., BYRD, R. J., RIZK, O. A. (1988) Creating and querying lexical databases. *Proceedings of the Association for Computational Linguistics Second Applied Conference on Natural Language Processing.* Austin, Texas, 84-92.

PISTOR, P., TRAUNMUELLER, R. (1986). A database language for sets, lists and tables. *Information Systems,* 11:4, 323-336.

ROTH, M. A., KORTH, H. F., SILBERSCHATZ, A. (1988). Extended algebra and calculus for nested relational databases. *ACM TODS,* 13:4.

SLATOR, B.M., WILKS. Y. (1987). Towards Semantic Structures for Dictionary Entries. *Proceedings of the 1987 Rocky Mountain Conference on Artificial Intelligence,* Boulder, CO, 85-98.

SPERBERG-MCQUEEN, M., BURNARD, L. (1990). Guidelines for the encoding and interchange of machine-readable texts, Draft, Version 0.0. ACH, ACL, and ALLC.

TOMPA, F. W. (1989). What is tagged text? *Proceedings of the 5th Annual Conference of the UW Centre for the New Oxford English Dictionary,* Oxford, 81-93.

VÉRONIS, J., IDE, N., M. (1990). Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries. *Proceedings of the 13th International Conference on Computational Linguistics,* COLING'90, Helsinki, 2, 389-394.

VÉRONIS J., WURBEL, N., HARIÉ, S., IDE, N.M. (1990). Contruction et exploitation d'une base de données lexicale multi-dictionnaires. *10th International Workshop on Expert Systems and their Applications,* Avignon, 85-104.